マルチ・スレッド・アンローダー

バージョン4.2.3

取扱説明書

2015年3月31日

株式会社プラムシックス

目次

表の索引

1. はじめに

この度は「マルチ・スレッド・アンローダー（英文製品名:Multi−threaded Unloader, 通称:MTU）」をご利用下さいまして誠に有難うございます。

ご使用の前に本書の内容をよく読まれた上で、正しくお使い下さいますよう謹んでお願い申し上げます。

サンプル版をお使いになる上での制限については本書の「6.3.. サンプル版の制限事項（32 頁）」をご覧下さい。

なお、旧バージョンからこのバージョンへ移行されるお客様はこの文書よりも前にリリースノート(release.txt)をお読みください。リリース・ノートでは次の点について解説があります。

- 旧バージョンから追加された機能
- バグの修正内容
- 新バージョンから廃止された機能
- 廃止になったパラメータ、環境変数
- 未修正のバグに対する暫定的な対処方法
- 将来廃止予定の現行機能

2. 使用上の注意

製品が出力するスクリプトを本番環境へ適用する際には特にご注意下さい。例えば UnLoad 機能や Query 機能（製品機能の名称）では SQL*Loader を内部的に実行するスクリプトで表の truncate table を実行します。あるいは GetMeta 機能では各種オブジェクトを create table する前に drop table を実行します。

truncate table および drop table といった DDL は永続的な操作です。その為、DDL を実行した後で間違いに気付いても回復が難しいか、あるいは事実上不可能な場合があります。

テキストデータの再ロード中は暗黙的に表ロックを獲得するので他のトランザクション処理がロック待機で長時間待たされる事が有ります。

上記に説明した動作が製品に期待する動作の一部である事をテスト環境などで十分確認してからお使い下さい。

また、この取扱説明書に記載された内容、および製品の仕様は、予告無く変更される場合があります。予めご了承下さい。

3. 追加された新機能

新しいバージョンでは（v4.2）より使い易さと性能を求める声に応えられるよう、多くの機能が追加されました。

この章ではこのバージョンで新しく追加された機能の概要を示します。更に詳しい記事は付属のリリースノート（release.txt）に掲載されていますので併せてご覧下さい。

表 1 新機能のまとめ

No	内容	カテゴリ
1.	製品のインストーラーを MSI (Microsoft Software Install) パッケージファイル形式で頒布するように改めました。Windows OS が備えているパッケージ管理機能を応用する事により製品のインストール状態を常に正しく維持出来るようになりました。 ● 参照先：インストール（39 頁） v3.06 以下で提供していた ZIP 形式書庫ファイルでのご提供はこの版以降で廃止されました。	全般
2.	x64 ベースの Windows へ対応する為、x64 ベースの実行形式ファイルを追加しました。従来は、x86 ベースの実行形式ファイルのみ提供していたので、x64 ベースの Windows 上で製品を使用する場合、Oracle Client for Windows (32-bit)を追加でインストールしなければなりませんでした。 もちろん、このバージョンに於いても従来通り WOW64 を利用した x86 ベースの実行形式ファイルを使用することが可能です。詳しい方法は「14..x64 ベースの Windows 上での使用（126 頁）」に書かれた記事を参照してください。	
3.	透過的アプリケーション・フェイルオーバー（TAF）へ対応しました。セッションが確立されると製品オリジナルの TAF イベントハンドラが OCI へフックされます。 この為 Oracle Real Application Cluster (RAC)、Oracle Fail Safe、Data Guard、アドバンスド・レプリケーション等の高信頼性オプションを有効に機能させることができます。 ● 参照先：17.TAF イベント対応（136 頁）	
4.	パイプを使ったプロセス間通信(IPC)を並列処理できるようになりました。標準入力からテキストデータを受取れるように設計された任意のコンソール・アプリケーションを、MTU の子プロセスとして複数起動させ、それぞれの標準入力に対して名無しパイプを経由して MTU から同時にデータを送り込むことができます。 パイプを経由すると中間ファイルが要らなくなる為、待機時間の多くを費やす I/O 待機が無くなりディスクの消費を抑えることもできます。 また、IPC で接続されたプロセスは個々に CPU 時間与えられて並列に動作するので Many Core システムの膨大な CPU 資源を集中的に投入して処理時間の短縮を図ることが可能になります。 ● 参照先：19.STREAM_LOCATOR の使用方法（142 頁）	UnLoad と Query
5.	名前付きパイプへデータを出力できるようになりました。効果という点では前述の名無しパイプと同じですが、名前付きパイプはファイル名の代わりに扱うことができる為、汎用性がより高い利用方法を選択することが可能になりました。 ファイルからテキストデータを入力可能なあらゆる既存のユーザ・プログラム資産をそのまま MTU との IPC の相手方として流用することが可能です。ユーザ・プログラム側では今まで入力用のファイル名を記述していた箇所に名前付きパイプの名前を代わりに記述することができます。 更にユーザ・プログラムがマルチスレッド・プロセスとして設計されていれば、データ・フローの上流から下流に至るまでのすべての経路をパイプライン化することが可能になります。 ● 参照先：19.STREAM_LOCATOR の使用方法（142 頁）	

No	内容	カテゴリ
6.	ディスクの空き領域を監視しながらデータを出力することが可能になしました。今までは、ディスクが一杯になるまで書込み続ける仕様だったのでディスクの空き領不足が深刻な事態を招く他のプログラムとの共存に問題がありました。 環境変数設定によりディスクの空き領域不足に対して2つの対応方針を事前に与えることが可能になりました。 1. 空き領域の閾値を設定して、その閾値を下回った時に事前に決めておいた1つ以上の代替場所へ出力先を切り替える(フェイル・オーバー)。 2. すべての代替場所の空き領域が閾値以下になったら自発的に停止する(フェイル・セーフ)。 ● 参照先:18.領域あふれ時のフェイルオーバー(139 頁)	
7.	メイン・フレーム向けのデータ移行が可能になりました。Oracle データベースのデータをメインフレームが直接受取ることができる「SAM((Sequential Access Method;順編成法)」と呼ばれる形式のファイルへ変換して出力することが可能です。出力形式について下記のようなオプションを選択することができます。 • NCHAR, NVARCHAR2 型を UTF-16BE で出力 • NUMBER 型をパック 10 進数 (COMP-3),DISPLAY SIGN LEADING SEPARATE で出力 • 改行なしの固定長データ形式で出力 ● 参照先:22.メインフレーム向けデータ移行(154 頁)	
8.	CreateFile (Windows API)の FILE_FLAG_NO_BUFFERING をサポートすることにより、データ出力先として使用しているディスク・デバイスの最適な I/O サイズを選択することが可能になりました。 弊社内のテストでは環境変数 OVERLAP_BUFFER_LENGTH(89 頁)をディスクの最適 I/O サイズへチューニングすると、旧版 v3.063 と比較して 19.6%性能を向上させることが可能であることを確認しました。 シリアル・アクセスのスループットが 100MB／秒を超える高性能ディスク・デバイスの潜在能力を引き出す事が可能になりました。 従来は Windows が管理するバッファサイズ(64kB)で規定される I/O サイズしか選べず、IOPS の上限によりボトルネックが生じ易かった点を改良しました。	
9.	並列処理データの単一ファイル出力が可能になりました。従来はパーティション表の並列出力結果はパーティションごとにファイルが分かれてしまう為、既存のプログラム資産の入力として使う場合に扱いにくさがありました。 この機能はパーティション表の並列アンロードだけでなくヒープ構成表の並列アンロードや集合化クエリという機能と組み合わせて使用することができます。 ● 参照先:環境変数 PARTITIONING(89 頁)	
10.	LOB の移動を快適するためのオプションが追加されました。このオプションを有効化すると LOB が表に大量に含まれる場合であっても、LOB が列毎にひとつのファイル(セカンダリ・データ・ファイル)へまとめて出力されるので、ファイルのオープン／クローズに要するオーバーヘッドやカーネル・メモリの使用量が大幅に削減されます。 ● 参照先:環境変数 MERGE_LOBS_INTO_SDF(85 頁)	
11.	ヒープ構成表の並列アンロードが可能になりました。ヒープ構成表を ROWID をベースとする複数の範囲に分割してそれぞれの範囲から(あたかもパーティション表のように)並列してデータをアンロードすることができます。 従来はヒープ構成表をアンロードするのに1つのスレッドしか割り当てられなかった為、巨大なヒープ構成表が1つだけあるようなスキーマのアンロードでは単一 CPU コアの能力に制限されてマルチスレッド化しても性能向上の効果がありませんでした。 ● 参照先:20.ヒープ構成表の並列アンロード(150 頁)	UnLoad のみ

No	内容	カテゴリ
12.	”集合化クエリ”（製品における造語）が使えるようになりました。集合化クエリは2つ以上の要素問合せ（SELECT 文）を含むスクリプト・ファイルで、MTU は集合化クエリが与えられると、記述された複数の SELECT 文の実行を同時に行います。 ● 参照先：21.集合化クエリ（152 頁）	Query のみ

4. 機能

4.1. 表データ･ファイル出力機能

（以下「UnLoad 機能」と表記）

Oracle の表に格納されたデータをCSV（カンマ区切りデータ）形式または固定長データ形式ファイルへ出力します。CSVの場合カンマ以外の任意の文字や文字列を区切り文字（文字列）として指定することも出来ます。
環境変数 TERMINATOR（98 頁）、EMBED_COLUMN_NAME（78 頁）、ENCLOSER, ENCLOSER2（78 頁）

- 各種データ型への対応は下記の通りです。

  網掛け表示したデータ型は環境変数 CHARSETID（75 頁）の設定内容の影響を受けます。例えば CHARSETID へ 873/871 を指定するとデータ･ファイルは UTF-8 で出力されます。または 0 を指定すると環境変数 NLS_LANG に指定されたキャラクタ･セットか、OS のデフォルト[1]の文字エンコーディングで出力されます。

表 2 UnLoad 可能な Oracle データ型

対応済	VARCHAR2	NVARCHAR2	NUMBER (FLOAT)	LONG
	DATE	CHAR	RAW	LONG RAW
	ROWID	NCHAR	BINARY_FLOAT	BINARY_DOUBLE
	CLOB	NCLOB[2]	BLOB	BFILE
	TIMESTAMP	TIMESTAMP WITH TIME ZONE	TIMESTAMP WITH LOCAL TIME ZONE	
	INTERVAL YEAR TO MONTH		INTERVAL DAY TO SECOND	
未対応	ユーザー定義型（オブジェクト型、VARRAY、Nested Table）		REF	UROWID

- SQL*Loader を使って元の表へデータ･ファイルをロードする場合に編集不要でそのまま利用可能な制御ファイルを出力します。「29.. 出力サンプル（186 頁）」に製品が出力する制御ファイルとデータ･ファイルのサンプルがありますのでご参照下さい。
- 出力対象表の主キー／一意キーを外部キーとして参照する表の参照制約、あるいは出力対象の表自身に施行されている参照制約を無効にしたり、有効にしたりする為のスクリプトを出力します。参照制約が有効のまま参照先主キー／一意キーを持つ表に対して、SQL*Loader を使って TRUNCATE/REPLACE メソッドにてデータのロードを行うと、下記の事例のエラーで失敗します。

```
SQL*Loader-926: executing delete/truncate (due to REPLACE/TRUNCATE keyword)
(表"DEPT")で OCI エラーが発生しました
ORA-02266: 表には有効な外部キーによって参照される一意キー/主キーが含まれています。
```

  製品が出力するスクリプトを事前に適用することでこのエラーを回避することができます。

- 出力対象の表の主キー制約･一意性制約を無効にしたり、有効にしたりする為のスクリプトを出力します。
- 出力対象の表の検査制約を無効にしたり、有効にしたりする為のスクリプトを出力します。
- 出力対象の表の Not Null 制約を無効にしたり、有効にしたりする為のスクリプトを出力します。

---

[1] 日本語 Windows の場合 Shift-SJIS となります

[2] NCLOB 型データに対するファイルはクライアントのキャラクタ･セットに関係なく、接続先データベースの各国語キャラクタ･セットに依存したエンコーディング方式で出力されます。通常は AL16UTF16 が選択されますので、UTF-16 Big Endian 方式のエンコーディングで出力されます。内容を確認するにはこのエンコーディング方式に対応したテキストエディタが必要です。

- 出力対象の表のデータベース･トリガーを無効にしたり、有効にしたりする為のスクリプトを出力します。
- 出力対象の表の索引を削除したり、再作成したりする為のスクリプトを出力します。
- 上記の作業で得たデータ･ファイルのロードやスクリプトの適用を一括処理するためのシェル･スクリプトを出力します。

  環境変数 MNGSCROUT （86 頁）、SKIPSCR（96 頁）

4.2. 任意問合せファイル出力機能

（以下「Query 機能」と表記）

- 任意問合せ（SELECT 文）を含むスクリプト･ファイルを予め作成しておき、この問合せを実行することによって戻される結果をCSVまたは固定長データ形式ファイルへ出力します。CSVの場合カンマ以外の任意の文字や文字列を区切り文字（文字列）として指定することも出来ます。

  環境変数 QUERYVAR（92 頁）、QUERYFIX（91 頁）、EXTNAMESQL（80 頁）
- 任意問合せをスクリプトファイルからではなく、環境変数の値として直接設定することが出来ます。

  環境変数 QUERYVAR（92 頁）、QUERYFIX（91 頁）
- 任意問合せにはシノニム、ビュー、マテリアライズド･ビュー及び分散データベースのリモート表への問い合わせや、結合文、福問合せを持つ文、WHERE 句、START WITH 句、CONNECT BY 句、GROUP BY 句、ORDER BY 句、及び HAVING 句を含むことが出来ます。
- 任意問合せに含まれるプレースフォルダへ値をバインドさせることにより、任意問合せが記述されたファイルを変更することなく選択条件を都度変えることが出来ます。

  環境変数 INPUT（83 頁）, FILEBIND（80 頁）
- バインド値をファイルからではなく、環境変数の値として直接 key=value の形式を設定することが出来ます。

  環境変数 FILEBIND（80 頁）
- SQL*Loader を使って任意の表へデータ･ファイルをロードする場合にそのまま編集不要で利用可能な制御ファイルを出力します。
- 上記の作業で得たデータ･ファイルのロードを一括処理するためのシェル･スクリプトを出力します。

  環境変数 SKIPSCR（96 頁）

4.3. パラレル再ロード処理機能

（以下「ReLoad 機能」と表記）

- 製品パッケージに含まれる別の実行形式ファイル(MPX.exe)がサポートする機能を使うことにより UnLoad や Query で得た SQL*Loader、SQL*Plus 用の入力ファイルを指定した並列度で再ロードすることが出来ます。
- サーバー、クライアント双方のアイドル待機時間を削減出来るため、再ロード所要時間を短縮することが可能になります。

  環境変数 PARALLELISM（90 頁）

4.4. スキーマオブジェクト定義 SQL 文出力機能

（以下「GetMeta 機能」と表記）

- 下記の表に示したオブジェクトの種類の定義文を出力します。

表 3 GetMeta がサポートするオブジェクトの種類

オブジェクトの種類	対応する文
プライベート・データベース・リンク	CREATE DATABASE LINK
シーケンス（順序）オブジェクト	CREATE SEQUENCE
索引クラスタ、ハッシュ・クラスタ	CREATE CLUSTER
クラスタ索引	CREATE INDEX … CLUSTER
ヒープ構成表（通常の表）,クラスタ表、パーティション表、外部表、索引構成表	CREATE TABLE
整合性制約	ADD CONSTRAINT … PRIMARY KEY ADD CONSTRAINT … UNIQUE ADD CONSTRAINT … FOREIGN KEY … REFERENCES ADD CONSTRAINT … CHECK
表レベルのサプリメンタル・ロギング	SUPPLEMENTAL LOG DATA SUPPLEMENTAL LOG GROUP
データベース・トリガー	CREATE TRIGGER
マテリアライズド・ビュー・ログ	CREATE MATERIALIZED VIEW LOG ON
B*木索引、ビットマップ索引、逆キー索引、ファンクション索引、パーティション索引	CREATE INDEX
LOB セグメント属性	LOB (…) STORE AS (…)
ビュー	CREATE VIEW
マテリアライズド・ビュー	CREATE MATERIALIZED VIEW
表コメント	COMMENT ON TABLE … ON
列コメント	COMMENT ON COLUMN … ON
プライベート・シノニム	CREATE SYNONYM
リフレッシュ・グループ、ジョブキューの定義文	DBMS_REFRESH.MAKE、DBMS_JOB.SUBMIT への呼び出しを含む無名 PL/SQL ブロック
ストアドプロシージャ、ストアドファンクション、パッケージ、及びタイプ	CREATE PROCEDURE CREATE FUNCTION CREATE PACKAGE CREATE PACKAGE BODY CREATE TYPE CREATE TYPE BODY
オブジェクト権限	GRANT … ON … TO …
標準監査	AUDIT … ON … BY …

- スキーマオブジェクトの定義文はオブジェクト毎に異なるファイルへ分割することも、一括して一つのファイルへ出力することも出来ます。

  環境変数 FILEMETA（81 頁）, METADIV（85 頁）

4.5. その他の機能

4.5.1. クイック・エクスポート機能

クイック・エクスポート機能は「4.2..任意問合せファイル出力機能（16 頁）」のパフォーマンスと柔軟性を手軽に利用できるようにするためのショートカットです。

環境変数を一つひとつ設定した上で実行しなければならない Query の煩わしさの代わりに、使い勝手の良い対話的なユーザインタフェースを提供します。

また、出力結果のデータ・ファイルは Microsoft® Excel®で扱うのに都合のよいデータ表現になっていますので、出力したデータを Excel で使うことが最初から決まっている場合にはこちらの利用が便利です。

ユーザインタフェースには下記の２つがあります。それぞれ、状況に合うものを選択する事ができます。

表 4:クイック・エクスポートについて

対話的ユーザインタフェースを提供するファイル名	説明
MTU.xla	Microsoft ® Excel® へ組み込んで使うための、アドイン・ファイルです。このファイルがアドインされると「マルチ・スレッド・アンローダー」というメニューバーが表示され、その中のメニューアイテムを選ぶと Query を実行するのに必要な値の入力をするのに便利なダイアログボックスが表示されます。 Query した時のログなども表示する機能があります。
QikExp.bat	Windows®のコマンドプロンプトから実行するためのバッチファイルです。実行すると Query を実行するのに必要な値の入力を促すプロンプトが幾つか表示され、入力項目が全部満たされると Query が開始されます。 一度入力した値は、次の実行時にデフォルト値として使えるよう保存する事もできます。

4.5.2. 整合性制約の無効化／有効化切替機能

（以下「ConsDisable/ConsEnable 機能」と表記）

- 指定したスキーマ内に含まれる有効な主キー制約、一意性制約、参照制約、検査制約、データベース・トリガーを一括して無効にします。
- 指定したスキーマ内に含まれる無効な主キー制約、一意性制約、参照制約、検査制約、データベース・トリガーを一括して有効にします。有効化する時に整合性制約違反の出た行に関する情報を EXCEPTIONS 表へ格納することが出来ます。

4.5.3. 外部キー再構成スクリプトファイル出力機能

（以下「GetFkrb 機能」と表記）

- CASCADE CONSTRAINTS オプション付の DROP TABLE 文を実行したときに削除される子表の参照制約を復元する為の定義 SQL 文を出力します。

  この機能は過剰に大きくなってしまった表の初期エクステントサイズを縮小する際の手順の一部を省力化することが出来ます。

4.5.4. 表間参照階層図（正展開）出力機能

（以下「GetTblP 機能」と表記）

- ある親テーブルを参照する全ての子テーブルの名前を再帰的に探し出し、テーブル名を参照階層のレベルに応じて字下げして出力します。

4.5.5. 表間参照階層図（逆展開）出力機能

（以下「GetTblN 機能」と表記）

- ある子テーブルによって参照される全ての親テーブルの名前を再帰的に探し出し、テーブル名を参照階層のレベルに応じて字下げして出力します。

4.5.6. 名前付き PL/SQL 間参照階層図（正展開）出力機能

（以下「GetPlsP 機能」と表記）

- ある名前付き親 PL/SQL（プロシージャ、ファンクション、パッケージ、タイプ）を参照する全ての名前付き子 PL/SQL の名前を再帰的に探し出し、オブジェクト名を参照階層のレベルに応じて字下げして出力します。

4.5.7. 名前付き PL/SQL 間参照階層図（逆展開）出力機能

（以下「GetPlsN 機能」と表記）

- ある名前付き子 PL/SQL（プロシージャ、ファンクション、パッケージ、タイプ）から参照される全ての名前付き親 PL/SQL の名前を再帰的に探し出し、オブジェクト名を参照階層のレベルに応じて字下げして出力します。

5. 特長

5.1. 全般

- Oracle Call Interface(Oracle 10gR2/11g/12c OCI)関数を C 言語から直接呼び出し、バルク操作で戻された行を一括処理する為高速に動作します。
- MTU はクライアント・プロセスとして作られているので、動作させるコンピュータや製品プログラムを実行するための OS アカウント、ファイル出力先の選択が自由です。

  これとは対照的に Data Pump Export/Data Pump Import ではサーバー・プロセスが出力ファイルを所有します。従って ORACLE.EXE(サーバープロセスの実行形式ファイル)を動かすローカル・コンピュータや OS アカウント(通常は LocalSystem)に与えられた権限による束縛を受けやすく使途の自由度が制限されます。
- (GUI を持たない)コンソール・アプリケーションとしてビルドされているので、お客様が作成されたバッチ処理などから呼び出して使うことが可能です。主要なスケジューリング用のミドルウエア上での実績も豊富です。

5.2. UnLoad 機能

- マルチスレッド・プロセスによる並列的なデータ読み取りをサポートします。ネットワーク通信やファイル操作の完了待ちでスレッドがブロックされている時間、他のスレッドへ CPU を割当てることが可能となり処理全体が高速化します。

  この特長は Oracle Database 10g 未満のバージョンか、Oracle Database 10g Standard Edition または Oracle Database 10g Standard Edition One をお使いの場合に重要です。これらの旧バージョンやエディションでは、Data Pump Export/Data Pump Import が存在しないかまたはパラレル操作が制限されているので、大量データの効率的なアンロード/再ロードの手段がありません。
  MTU はバージョンやエディションの違いによる並列化の制限を受けない為、ライセンスされた CPU コア数の上限、ディスク I/O 率、ネットワーク I/O 率などのシステム構成上のボトルネックのみにパフォーマンス上の制限が依存します。この制限は多くのユーザにとって容認できるものです。
  環境変数 PARALLELISM(90 頁)
- データの読取一貫性はマルチバージョン同時実行性制御(MVCC)により保たれます。他社の高速アンロードユーティリティーにあるような読取表への共有表ロックを必要としないのでアンロード中であってもトランザクションの実行が妨げられることはありません。その反対に、他にトランザクション処理が行われていても MTU は読取表へのロックを掛けないのでアンロードが妨げられることもありません。

  環境変数 CONSISTENT(76 頁)
- ASM(Automatic Storage Management)、RAW パーティション、あるいは OCFS 上に格納されたデータでも制限無くアンロードすることが出来ます。Oracle サーバー上で動作させることを強要されず、Oracle クライアントで動作させて接続時フェイルオーバさせることも出来ますので、RAC(Real Application Cluster)での運用に適しています。
- Unicode を使用するデータ・ファイル出力に対応します。接続先データベースに各言語のマルチバイト文字が含まれていても、文字化けすることなくアンロードできます。Unicode 出力対象の詳しい内容については「13..Unicode 対応について (125頁)」を参照してください。

  環境変数 CHARSETID(75 頁)
- 非同期ファイル操作をサポートします。ファイル操作完了をスレッドが待機しにくくなる為、少ないスレッド数でも実質的な並列度を上げる事ができます。並列度数を増やすと同時に増えてしまいがちなメモリ消費を抑えることができます。また **FILE_FLAG_NO_BUFFERING** を指定することにより、システムが用意するバッファキャッシュをバイパスできる効率の良いデータ書き込みを実現しました。

  環境変数 IO_OVERLAP_SCALE(84 頁)

- パーティション表に対応します。パーティション、またはサブパーティションごとに個別のスレッドを割当てデータ・ファイル作成を並列化できるので所要時間を短縮することができます。

  環境変数 PARTITIONING（89 頁）

- データの出力先を標準出力へ切り替えることが出来ます。標準入力からデータを取り込むことの出来る他のプログラムに対してパイプを使ってデータを提供することが出来ます。データのサイズが大きくなる場合、ディスクアクセスに伴う待機時間や空き領域を気にしなくて済みます。

  MTU とパイプでつながれた他のプログラムは並列に実行されるので CPU に余裕がある場合は所要時間が短縮される場合もあります。
  環境変数 STDOUT（97 頁）

- 任意のアプリケーション・コンテキストをデータ読取前に設定することができます。Virtual Private Database ではファイングレイン・アクセス・コントロールとアプリケーション・コンテキストを組み合わせてアプリケーションへ提供するデータのサブセットを動的に決める為、認証後の PL/SQL（プロシージャ：DBMS_SESSION.SET_CONTEXT）を使った操作を必要とします。

  環境変数 PRE_REP_EXEC_PLS（91 頁）、POST_REP_EXEC_PLS（90 頁）

- データ・ファイルの格納先を1つ以上の場所へ分散することが出来ます。分散した先が、ディスク回転軸を共有しない、独立して動作可能なデバイス上に設けられたファイルシステムの場合、アンロード時、再ロード時いずれの場合でも、I/O の競合が抑えられ所要時間の短縮を図ることが可能です。

  環境変数 AUXILIARY_DEST（74 頁）

- 権限の設定次第[3]で代理ユーザが他ユーザ所有の表データを出力することが出来ます。例えばアカウントロックされているスキーマに含まれるデータを、DBAが代行してアンロード・再ロードすることも出来ます。セキュリティー上の理由でアカウントをロックしユーザのログインを認めないような運用を行う場合にこの機能は役立ちます。

  環境変数 SRC_USER（97 頁）, ALTER_DEST（74 頁）

- ワイルドカード文字、およびカンマ区切リストを含む文字列を環境変数へ設定することにより、出力対象表と出力除外表を複数選択出来ます。

  環境変数 LISTTABLE（84 頁）, LISTEXCPT（84 頁）

- 出力対象表、出力除外表が多数に及ぶ場合、表名のリストをテキストファイルから取り込むことが出来ます。

  環境変数 FILETABLE（82 頁）, FILEEXCPT（81 頁）

- 固定長データ形式で出力したほうがアンロードや再ロードのパフォーマンスが向上させる事が出来る表を LISTTABLE、FILETABLE の中から選択できます。選択の際には、ワイルドカード文字や、カンマ区切りリストによる環境変数の設定、及び表名のリストをテキストファイルから取り込む事が出来ます。

  環境変数 LISTFIXED（84 頁）、FILEFIXED（81 頁）

- 表に N 行以上行が存在しない場合は、出力対象の表から除外するというルールを設定することが出来ます。

  環境変数 NUM_ROWS（88 頁）

- 一般には SQL*Loader 実行前に表に付属する整合性制約、トリガー、及び索引を外しておくと、ロード時のパフォーマンスが向上すると言われています。UnLoad ではこのような管理的な作業を支援するためのスクリプトを自動的に生成することが出来ます。

  環境変数 MNGSCROUT（86 頁）, SKIPSCR（96 頁）

- 主キー制約、または一意性制約が施行されている表はキー昇順に並び替えてデータを出力出来ます。一意索

[3] 次の何れかの追加作業により対応可能です。
1.GRANT ANY ROLE または GRANT ANY PRIVILEGE システム権限を持つユーザ（SYS/SYSTEM 等）が実行する。
1-1.代理ユーザに対して(a)のシステム権限か、この権限を含むロールを与える。
1-2.PUBLIC ロールに対して(a)のシステム権限か、この権限を含むロールを与える。
2.表の所有者が実行する。
2-1.代理ユーザに対して表の個々に関する(b)のオブジェクト権限か、この権限を含むロールを与える。
2-2.PUBLIC ロールに対して表の個々に関する(b)のオブジェクト権限か、この権限を含むロールを与える。

引のクラスタリング・ファクタを改善（最小化）する時に役立ちます。あるいは、diff(UNIX の場合)や fc(Windows の場合)などの外部コマンドを使ってデータ・ファイル同士の差異を調べることも出来ます。

ただし大きな表のデータ・ファイルを出力する場合は、並び替えの為のワークとして行数に応じたサイズの一時セグメントと追加の所要時間を必要とする場合があります。並び替えが必要でない場合は、実行時のオプションで無効にすることが出来ます。
環境変数 SORTING（96 頁）, SORT_AREA_SIZE（96 頁）

- OS ファイルシステムに最大ファイルサイズの制限がある場合、その制限値を超えないサイズに分割された複数のデータ・ファイルを作成することが出来ます。この時、制御ファイルの INFILE 句には分割された全てのファイル名が参照されるように記述されます。

  環境変数 FILESIZE（82 頁）

- Oracle 8 から採用された新しいデータ型である、CLOB 型、NCLOB 型、及び BLOB 型の LOB 値を個別のファイル（LOBFILE）か、または列毎に LOB をまとめて単一ファイル（SDF）として出力できます。

  Oracle 10g 以上のクライアントを使っている場合には 4GB を超える LOB ファイルを出力する事ができます。
  LOBFILE を選択した場合、個々の LOB 出力が並列で実行されます。
  また BFILE 型のディレクトリ別名とファイル名を出力できます[4]。さらに NCHAR 型、NVARCHAR2 型のデータも出力できます。

- v4.18 から CLOB/NCLOB のアンロードには新しくセカンダリ・データ・ファイル（SDF）が選択できるようになりました。SDF へ LOB データを出力すると LOB データが列ごとにひとつのファイルへまとめられる為、従来の LOBFILE と比べると出力されるファイルの数を大きく削減することが可能になります。結果的にはこれが、アンロードとリロードの性能向上に大きく寄与する事になりました。

  環境変数 MERGE_LOBS_INTO_SDF (85頁)

- Oracle9i から採用された新しいデータ型である、TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE,INTERVAL YEAR TO MONTH, INTERVAL DAY TO SECOND を含む表のデータを出力できます。

- Oracle10g から採用された新しいデータ型である、BINARY_FLOAT, BINARY_DOUBLE を含む表のデータを出力できます。

- 列値に非表示文字や改行文字（LF）が含まれていても SQL*Loader が正しく取り込めるような形式でデータを出力できます。

  環境変数 RECLENGTH（93 頁）

- 最大 2GB（2 の 31 乗から 1 を引いたバイト数）の可変長の文字列及びバイト列が格納された LONG 型及び LONG RAW 型のデータを出力できます（ただしメモリが十分に搭載されている前提で）。

  環境変数 MAXLONGSIZE（85 頁）

- LONG 型及び LONG RAW 型が表に含まれる場合、SQL*Loader 使用時に CLOB 型,BLOB 型,BFILE 型として格納できるデータ表現へ変換することが出来ます。言換えると CLOB 型,BLOB 型,BFILE 型を読み込んだのと同じ結果になる様に出力することが出来ます。この機能を使用すれば LONG 型および LONG RAW 型をより制限の少ないオラクルの新しいデータ型へ速やかに移行することが出来ます[5]。

  環境変数 LONGTRANSIT（85 頁）

- 外部表を定義する為のスクリプト(ORGANIZATION EXTERNAL オプション付きの CREATE TABLE 文)を出力することが出来ます。外部表を定義すると CSV や固定長テキストファイルに対して直接 SELECT 文を実行することが出来るようになります。

  環境変数 GETEXTTBL（83 頁）

- フラッシュバック問合せへの対応により過去の指定した時点に存在していたデータを出力することができます。これによりユーザエラーによって失われた行データを含む一貫した表のイメージをフラッシュバック目標の SCN か日付時刻の何れかを与えることにより容易に復元できます。

  環境変数 FBQSCN（80 頁）, FBQTIME（80 頁）

---

[4] ただし BFILE ロケータが指し示すファイルの内容は出力できません。

[5] オラクル製品のリリース・ノートによれば将来、LONG 型、LONG RAW 型はサポートされなくなると明記されています。

- 出力対象テーブルのセグメントサイズをディクショナリから取得し、領域サイズの降順でアンロードします。テーブルが LOB を持っている場合、LOB セグメントの合計サイズも、領域サイズ比較に算入されます。これは下図のように、出力対象テーブル中に一際大きなサイズのテーブルが存在するとき、所要時間の短縮に効果があります。

5.3. GetMeta 機能

- データベース・オブジェクトの CREATE 文がデフォルトでは meta.sql という単一のファイルへ格納されますので、grep(UNIX の場合)や findstr(Windows の場合)などの外部コマンドを使ってオブジェクト名の探索をするのに便利です。

  環境変数 METADIV（85 頁）
- Oracle Database 11g 以下で使用できる全てのパーティション化方法に対応します。

表 5 パーティション化方法と登場時期

パーティション化方法				
	非コンポジット	コンポジット		
		レンジ	リスト	ハッシュ
レンジ	8	11g	9i	8i
リスト	9i	11g	11g	11g
ハッシュ	8i	×	×	×

インバーバル・パーティショニング	11g
リファレンス・パーティショニング	11g
バーチャルカラム・パーティショニング	11g

- TEMPLATE 句や STORE IN 句などの無名パーティション化表現を含むパーティション表、およびその LOCAL 索引などの CREATE 文出力では、オリジナルの SQL に近い表現が入手できます[6]。例えば、下記のようなを使った TEMPLATE 句を含むレンジ・ハッシュ・コンポジット・パーティション表、およびその LOCAL 索引を作成したとします。

[6] 無名である事を無名のまま表現することが出来ます。

```
create table TEST24 (
   aria_id        NUMBER (4)
   , mp_id          NUMBER (4)
   , item_id        NUMBER (4)
   , tagno          CHAR (10)
   , dt             DATE
   , value          NUMBER (12,3)
   , status         NUMBER (4)
   , constraint test24_pk primary key (aria_id, mp_id, item_id)
) storage (initial 128k next 128k minextents 1 maxextents 65536)
partition by range (dt)
subpartition by list (item_id)
subpartition template
( subpartition p1 values ('1','8','9') tablespace PAR1
, subpartition p2 values ('2','4','5','7') tablespace PAR2
, subpartition p3 values (NULL) tablespace PAR3
, subpartition p4 values (DEFAULT) tablespace PAR4
)
( partition p1 values less than (to_date('20050401','yyyymmdd'))
, partition p2 values less than (to_date('20050501','yyyymmdd'))
, partition p3 values less than (to_date('20050601','yyyymmdd'))
, partition p4 values less than (MAXVALUE)
)
;
create index TEST24_IE01 on TEST24 (item_id)
local;
```

DBMS_METADATA.GET_DDL パッケージ・プロシージャでこれを出力すると下記のような表現を得ます。この表現は情報量に洩れがない為、リカバリが目的の場合には使用可能です。
しかし TEMPLATE 句を使用してサブパーティションについては無名化を要求しているにも拘らず、「SUBPARTITION "P1_P1"」のように Oracle が自動的に生成したオブジェクト名とその機械的な繰り返しに還元されてしまい、オリジナルの SQL で表現されていた抽象性のレベルが損なわれてしまいます。
これではリバース・エンジニアリングの資源として扱う用途に適していません。

```
SQL> set pagesize 2000 long 10000 linesize 180
SQL> select dbms_metadata.get_ddl('TABLE', 'TEST24', 'SCOTT') as DDL from dual ;

DDL
--------------------------------------------------------------------------------


  CREATE TABLE "SCOTT"."TEST24"
   (    "ARIA_ID" NUMBER(4,0),
        "MP_ID" NUMBER(4,0),
        "ITEM_ID" NUMBER(4,0),
        "TAGNO" CHAR(10),
        "DT" DATE,
        "VALUE" NUMBER(12,3),
        "STATUS" NUMBER(4,0),
         CONSTRAINT "TEST24_PK" PRIMARY KEY ("ARIA_ID", "MP_ID", "ITEM_ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 131072 NEXT 131072 MINEXTENTS 1 MAXEXTENTS 2147483645
```

```
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
  STORAGE(INITIAL 131072 NEXT 131072 MINEXTENTS 1 MAXEXTENTS 65536
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"
  PARTITION BY RANGE ("DT")
  SUBPARTITION BY LIST ("ITEM_ID")
  SUBPARTITION TEMPLATE (
    SUBPARTITION "P1" values ( '1', '8', '9' )
  TABLESPACE "PAR1" ,
    SUBPARTITION "P2" values ( '2', '4', '5', '7' )
  TABLESPACE "PAR2" ,
    SUBPARTITION "P3" values ( NULL )
  TABLESPACE "PAR3" ,
    SUBPARTITION "P4" values ( DEFAULT )
  TABLESPACE "PAR4"  )
 (PARTITION "P1"  VALUES LESS THAN (TO_DATE(' 2005-04-01 00:00:00', 'SYYYY-MM-DD
HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
  STORAGE(INITIAL 131072 NEXT 131072 MINEXTENTS 1 MAXEXTENTS 65536
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"
 ( SUBPARTITION "P1_P1"  VALUES ('1', '8', '9')
   TABLESPACE "PAR1",
  SUBPARTITION "P1_P2"  VALUES ('2', '4', '5', '7')
   TABLESPACE "PAR2",
  SUBPARTITION "P1_P3"  VALUES (NULL)
   TABLESPACE "PAR3",
  SUBPARTITION "P1_P4"  VALUES (DEFAULT)
   TABLESPACE "PAR4") ,
 PARTITION "P2"  VALUES LESS THAN (TO_DATE(' 2005-05-01 00:00:00', 'SYYYY-MM-DD
HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
  STORAGE(INITIAL 131072 NEXT 131072 MINEXTENTS 1 MAXEXTENTS 65536
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"
 ( SUBPARTITION "P2_P1"  VALUES ('1', '8', '9')
   TABLESPACE "PAR1",
  SUBPARTITION "P2_P2"  VALUES ('2', '4', '5', '7')
   TABLESPACE "PAR2",
  SUBPARTITION "P2_P3"  VALUES (NULL)
   TABLESPACE "PAR3",
  SUBPARTITION "P2_P4"  VALUES (DEFAULT)
   TABLESPACE "PAR4") ,
 PARTITION "P3"  VALUES LESS THAN (TO_DATE(' 2005-06-01 00:00:00', 'SYYYY-MM-DD
H24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
  STORAGE(INITIAL 131072 NEXT 131072 MINEXTENTS 1 MAXEXTENTS 65536
  BUFFER_POOL DEFAULT)
```

```
  TABLESPACE "USERS"
 ( SUBPARTITION "P3_P1"  VALUES ('1', '8', '9')
   TABLESPACE "PAR1",
  SUBPARTITION "P3_P2"  VALUES ('2', '4', '5', '7')
   TABLESPACE "PAR2",
  SUBPARTITION "P3_P3"  VALUES (NULL)
   TABLESPACE "PAR3",
  SUBPARTITION "P3_P4"  VALUES (DEFAULT)
   TABLESPACE "PAR4") ,
 PARTITION "P4"  VALUES LESS THAN (MAXVALUE)
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
  STORAGE(INITIAL 131072 NEXT 131072 MINEXTENTS 1 MAXEXTENTS 65536
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"
 ( SUBPARTITION "P4_P1"  VALUES ('1', '8', '9')
   TABLESPACE "PAR1",
  SUBPARTITION "P4_P2"  VALUES ('2', '4', '5', '7')
   TABLESPACE "PAR2",
  SUBPARTITION "P4_P3"  VALUES (NULL)
   TABLESPACE "PAR3",
  SUBPARTITION "P4_P4"  VALUES (DEFAULT)
   TABLESPACE "PAR4") )


SQL> select dbms_metadata.get_ddl('INDEX', 'TEST24_IE01', 'SCOTT') as DDL from dual ;

DDL
--------------------------------------------------------------------------------


  CREATE INDEX "SCOTT"."TEST24_IE01" ON "SCOTT"."TEST24" ("ITEM_ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(
  BUFFER_POOL DEFAULT) LOCAL
 (PARTITION "P1"
PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(
  BUFFER_POOL DEFAULT)
 ( SUBPARTITION "P1_P1"
   TABLESPACE "PAR1",
  SUBPARTITION "P1_P2"
   TABLESPACE "PAR2",
  SUBPARTITION "P1_P3"
   TABLESPACE "PAR3",
  SUBPARTITION "P1_P4"
   TABLESPACE "PAR4") ,
 PARTITION "P2"
PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(
  BUFFER_POOL DEFAULT)
 ( SUBPARTITION "P2_P1"
```

```
    TABLESPACE "PAR1",
   SUBPARTITION "P2_P2"
    TABLESPACE "PAR2",
   SUBPARTITION "P2_P3"
    TABLESPACE "PAR3",
   SUBPARTITION "P2_P4"
    TABLESPACE "PAR4") ,
  PARTITION "P3"
 PCTFREE 10 INITRANS 2 MAXTRANS 255
   STORAGE(
   BUFFER_POOL DEFAULT)
  ( SUBPARTITION "P3_P1"
    TABLESPACE "PAR1",
   SUBPARTITION "P3_P2"
    TABLESPACE "PAR2",
   SUBPARTITION "P3_P3"
    TABLESPACE "PAR3",
   SUBPARTITION "P3_P4"
    TABLESPACE "PAR4") ,
  PARTITION "P4"
 PCTFREE 10 INITRANS 2 MAXTRANS 255
   STORAGE(
   BUFFER_POOL DEFAULT)
  ( SUBPARTITION "P4_P1"
    TABLESPACE "PAR1",
   SUBPARTITION "P4_P2"
    TABLESPACE "PAR2",
   SUBPARTITION "P4_P3"
    TABLESPACE "PAR3",
   SUBPARTITION "P4_P4"
    TABLESPACE "PAR4") )
```

ところがMTUのGetMetaでは下記のような表現を得られるので、リバース・エンジニアリングが容易です。

```
/*
 * Tabs: 000018. TEST24
 */
drop table "TEST24";
create table "TEST24"
( /* 0001 */ "ARIA_ID"                     number(4)
, /* 0002 */ "MP_ID"                       number(4)
, /* 0003 */ "ITEM_ID"                     number(4)
, /* 0004 */ "TAGNO"                       char(10)
, /* 0005 */ "DT"                          date
, /* 0006 */ "VALUE"                       number(12,3)
, /* 0007 */ "STATUS"                      number(4)
, constraint "TEST24_PK"                   primary key
  ( /* 0001 */ "ARIA_ID"
  , /* 0002 */ "MP_ID"
```

```
  , /* 0003 */ "ITEM_ID"
  ) not deferrable initially immediate disable validate
) pctfree 10 pctused 40 initrans 1 maxtrans 255
  nocache parallel (degree 1 instances 1) tablespace "USERS"
  storage (initial 128k next 128k minextents 1 maxextents 65536
  buffer_pool DEFAULT)
partition by RANGE ("DT")
subpartition by LIST ("ITEM_ID")
subpartition template
( /* 0001 */ subpartition "P1" values ('1', '8', '9') tablespace "PAR1"
, /* 0002 */ subpartition "P2" values ('2', '4', '5', '7') tablespace "PAR2"
, /* 0003 */ subpartition "P3" values (null) tablespace "PAR3"
, /* 0004 */ subpartition "P4" values (default) tablespace "PAR4"
)
( /* 0001 */ partition "P1" values less than (to_date('2005-04-01','YYYY-MM-DD'))
, /* 0002 */ partition "P2" values less than (to_date('2005-05-01','YYYY-MM-DD'))
, /* 0003 */ partition "P3" values less than (to_date('2005-06-01','YYYY-MM-DD'))
, /* 0004 */ partition "P4" values less than (maxvalue)
) ;
create index "TEST24_IE01" on "TEST24"
( /* 0003 */ "ITEM_ID"
) pctfree 10 initrans 2 maxtrans 255
  parallel (degree 1 instances 1)
  storage (buffer_pool DEFAULT)
local
;
```

- 11g 以下でサポートされているすべての表圧縮に対応しています。
- 表レベルで設定されているサプリメンタル・ロギングに対応しています。
- 文監査オプションを有効にしておくと監査証跡との組合せにより、「いつ」「誰が」オブジェクトの「どこ」を「どのように」改変したかを監査することが可能になります。

  環境変数 NUMDAYS_AUDIT（88 頁）
  例えば下記のようなスクリプトを用意し、標準監査の開始と表 FOO 作成、表 FOO に対する幾つかの操作を実行したとします。

```
connect / as sysdba
alter system set audit_trail=DB scope=spfile;
shutdown immediate
startup
audit table, index, alter table by scott /* 文監査 */;
connect scott/tiger
create table FOO (c1 varchar2(3));
/* ここで数分間待機 */
alter table FOO add (c2 number(10,9));
alter table FOO add constraint FOO_PK primary key (c1);
/* ここで数分間待機 */
create index FOO_IE01 on FOO (c2);
exit
```

環境変数 NUMDAYS_AUDIT へ 1 以上の値をセットし GetMeta を実行すると表 FOO について下記のようなス

クリプトとコメントを得ることが出来ます。主キーは CREATE TABLE 文によって作成されたのではなく、ALTER TABLE 文によって後から追加されたことが監査証跡から分かります。同じように索引も CREATE TABLE とは別な時期に作成されたことが分かります。

```
/*
 * Tabs: 000004. FOO
 */
drop table "FOO";
create table "FOO"
( /* 0001 */ "C1"                          varchar2(3)
, /* 0002 */ "C2"                          number(10,9)
, constraint "FOO_PK"                      primary key
  ( /* 0001 */ "C1"
  ) not deferrable initially immediate disable validate
) pctfree 10 pctused 40 initrans 1 maxtrans 255
  nocache parallel (degree 1 instances 1) tablespace "SYSTEM"
  storage (initial 64k minextents 1 maxextents unlimited
  freelists 1 freelist groups 1 buffer_pool DEFAULT)
;
create index "FOO_IE01" on "FOO"
( /* 0002 */ "C2"
) pctfree 10 initrans 2 maxtrans 255
  parallel (degree 1 instances 1) tablespace "SYSTEM"
  storage (initial 64k minextents 1 maxextents unlimited
  freelists 1 freelist groups 1 buffer_pool DEFAULT)
;
create UNIQUE index "FOO_PK" on "FOO"
( /* 0001 */ "C1"
) pctfree 10 initrans 2 maxtrans 255
  parallel (degree 1 instances 1) tablespace "SYSTEM"
  storage (initial 64k minextents 1 maxextents unlimited
  freelists 1 freelist groups 1 buffer_pool DEFAULT)
;
alter table "FOO"
  enable validate constraint "FOO_PK"
;
/*
The audit trails of the target actions.
2008-09-06 22:35:19, 14859, 1, CREATE TABLE, FOO, hoge¥foo, bar, SCOTT, 0, CREATE TABLE
2008-09-06 22:39:03, 14859, 2, ALTER TABLE, FOO, hoge¥foo, bar, SCOTT, 0,
2008-09-06 22:39:03, 14859, 3, CREATE INDEX, FOO_PK, hoge¥foo, bar, SCOTT, 0,
2008-09-06 22:39:03, 14859, 4, ALTER TABLE, FOO, hoge¥foo, bar, SCOTT, 0,
2008-09-06 22:40:51, 14859, 5, CREATE INDEX, FOO_IE01, hoge¥foo, bar, SCOTT, 0,
*/
```

- 参照制約によって参照されている親表が子表よりも前方に定義されるように配置することが出来ます。従いまして作成されたスクリプトファイルをそのまま SQL*Plus で実行しても親表が存在しないことによる例外が発生しません。

  環境変数 DEPENDENT（77 頁）

- 参照関係にある名前付き PL/SQL（プロシージャ、ファンクション、パッケージ、タイプ、データベース・トリガー）においては名前付き親 PL/SQL が名前付き子 PL/SQL よりも前方に定義されるように配置されます。

- 表、索引、クラスタ、MVIEW, MVIEW LOG, オーバーフローセグメント、及び LOB セグメントについては複数の追加エクステントが発生している場合、初期エクステントへまとめるように値を調節された STORAGE 句オプション付きの定義 SQL 文を出力することが出来ます。

  環境変数 COMPRESS（76 頁）
- 権限の設定次第[7]で代理ユーザが他ユーザ所有のオブジェクトについて定義 SQL 文を出力することが出来ます。

[7] 全てを表示するのに必要な追加作業について説明致します。
1.GRANT ANY ROLE または GRANT ANY PRIVILEGE システム権限を持つユーザ（SYS/SYSTEM 等）が実行する。
1−1.代理ユーザに対して(c)のシステム権限か、この権限を含むロールを与える。
1−2.PUBLIC ロールに対して(c)のシステム権限か、この権限を含むロールを与える。
2.データベースオブジェクトの所有者が実行する。
2−1.代理ユーザに対してオブジェクトの個々に関する(d)のオブジェクト権限か、この権限を含むロールを与える。
2−2.PUBLIC ロールに対してオブジェクトの個々に関する(d)のオブジェクト権限か、この権限を含むロールを与える。
(a) select any table
(b) select
(c) select any table, select any sequence, comment any table, execute any procedure, create any trigger
(b) select, execute

6. 動作環境と制限

6.1. 必要システム構成

以下の条件を満たすシステムでご使用下さい。

表 6:必要システム構成

要求要素	条件
OS	Windows Vista Service Pack 2 Windows 7 Service Pack 1 Windows 8 Windows 8.1 Windows Server 2008 Service Pack 2 Windows Server 2008 R2 SP1 Windows Server 2012 Windows Server 2012 R2
CPU	1.6GHz または Intel® Pentium® 以上
メモリ	1GB 以上(の RAM
ディスク	10 GB 以上の固定ディスク空き領域
モニタ	表示解像度 1024×768 以上
補助記憶装置	インストール時のみ CD-ROM 必要

製品を使用するシステムには Microsoft 社が提供する「Visual Studio 2013 の Visual C++ 再頒布可能パッケージ」が予めインストールされている必要があります。

表 7:再頒布可能パッケージの入手先

パッケージ名	URL
Visual Studio 2013 の Visual C++ 再頒布可能パッケージ	http://www.microsoft.com/ja-jp/download/details.aspx?id=40784

- 上記 URL は当文書作成時点におけるものです。将来的に変更される可能性がございますのでご了承ください。
- 製品の頒布用 CD-R には上記のパッケージをダウンロードしたものが含まれていますのでご利用下さい。
- このパッケージのインストール方法については 30.Visual Studio 2013 の Visual C++ 再頒布可能 パッケージの導入(190 頁)にある記事を参照してください。

製品プログラムを動作させるためには、「表 8:MTU が動作する為に必要な Oracle 製品(32 頁)」に挙げた Oracle 製品の何れかがマニュアル通りに正しくセットアップされたコンピュータが必要です。

表 8:MTU が動作する為に必要な Oracle 製品

Oracle 製品の種類	バージョン	エディション
Oracle Database	10gR2, 11g, 12c	Enterprise Edition
		Standard Edition
	10gR2, 11g, 12c	Standard Edition One
		Personal Edition
Oracle Client	10gR2, 11g, 12c	
Oracle Instant Client	10gR2, 11g	

- 接続先 Oracle データベースは 8i (Release 8.1.7) 以上であれば、エディション・オプションの種類、OS の種類、bit 数、使用されるキャラクタ・セット等に制限はありません。但し、(Oracle Database 8i と Oracle 11g Client の組合せなど)オラクル社がサポートしていない組合せの場合は MTU もサポートいたしません。

  オラクル社がサポートする組み合わせについては My Oracle Support[8] のメニューにある「動作保証」でご確認ください(要アカウント)。
- x64 ベースの Windows 上で、製品に含まれる x86 ベースの実行形式ファイルを使用する場合は、WOW64 による 32-bit エミュレーター環境でのみ動作します。この場合、Oracle Database for Microsoft Windows (32-bit)か Oracle Database Client for Microsoft Windows (32-bit)の何れかが製品マニュアル通りに正しくインストールされている必要があります。WOW64 を使って MTU をこれらのプラットホームで動かす方法については「14.x64 ベースの Windows 上での使用(126 頁)」をご覧ください。
- 製品インストールとアンインストール時に限り、インストール対象コンピュータの管理者権限をもつユーザでのログインが必要となります。製品インストール後の通常運用では管理者権限は不要です。

6.2. 全般的な制限事項

- 同一 bit 数の製品パッケージを1台のコンピュータへ複数インストールすることはできません。これは、バージョンや版(有償版/サンプル版)の違いがあっても制限されます。

  64-bit 版、32-bit 版それぞれ1セットずつであれば、64-bti ベースの Windows 上に同居させることができます。
- Intel Itanium ベースの Windows へは未対応です。
- オブジェクト表、逆キー索引、ドメイン索引、XML DB へは未対応です。
- 名前付きデータ型には未対応です。
- 読取り専用モードでオープンされたスタンバイデータベースに対して、UnLoad 機能と GetMeta 機能を使用することはできません。

6.3. サンプル版の制限事項

- UnLoad/Query 時に 10 万行を超えるデータを戻すことができません。
- UnLoad 時に出力される＜表名＞_DEPS.sql ファイル中の create index 文に"tablespace <表領域名>" の表示が出力されません。
- GetMeta 時に "tablespace <表領域名>" と compress の表示が出力されません。

[8] https://support.oracle.com/

弊社ではサンプル版の他に、行数制限や表領域名表示の制限が無く、使用期間限定の評価版ライセンスを用意しております。詳しくは「33.製品に関する問合せ先(194 頁)」にてご確認下さい。

6.4. ライセンス認証の概要

- 製品の有償版 v4.1 以降でライセンス認証という機能が追加されました。ライセンス認証はカジュアルコピーを手にした使用者が不正に製品を使用できてしまう不公平を排除し、製品の正当な使用者たるお客様の権利を保護する為のものです。
- この為、ライセンス認証が完了しないと製品を使い始める事ができない制限が組み込まれています。
- 更にライセンス認証が完了した後に、同じライセンス証書のシリアル番号を使って別なコンピュータでライセンス認証を行おうとするとエラーとなり完了させることができません。
- ただし上記には例外があり、製品にバンドルされた「ライセンス認証登録の抹消」というツールを使用して認証登録情報を一旦取り消すと、別なコンピュータでのライセンス認証(=移動)が可能になります。
- また、「ライセンス認証登録の抹消」自体が利用できない緊急事態(たとえばハードウエアの故障)の場合は弊社のライセンス認証担当者へお電話を頂き事情をご説明いただいた上で、代替コンピュータでのライセンス認証が可能になるよう手続きをさせて頂くことが可能です。代替コンピュータの CPU コア数は、障害のあったコンピュータよりも少ないか、等しいものを選んでください。
- ライセンス認証の過程で製品プログラムは、お客様のご使用しているコンピュータのハードウエア情報の内、次のものをライセンス認証に必要な基礎情報として収集します。

  a. MACアドレス

  b. CPUのメーカ・型番・クロック周波数

  c. 全CPUコアの総数

  d. 固定ディスクのメーカ・型番・製造番号

- 製品プログラムは上記の基礎情報を収集すると、ハッシュアルゴリズムを使って4バイトのハッシュ値を作成します。このハッシュ値をライセンス証書の証書シリアル番号とともに開発元のライセンス認証サーバーへ送信します。

  製品プログラムは収集した基礎情報をそのまま送信するのではなく、ハッシュ値に変換したものを送信する点にご留意願います。ハッシュ値はその数学的性質により、多数のシノニムが存在するので変換元の基礎情報を特定することが技術的に困難です。また、基礎情報が1ビットでも異なれば、ハッシュ値も変わるので、ライセンス証書シリアル番号と認証されたハードウエアの関連付けを定義するには情報量として十分です。

- ライセンス証書の証書シリアル番号と認証されたハードウエアは1対1で対応します。従いまして、ひとつの証書シリアル番号を複数のハードウエアのライセンス認証に使用することができません。MTU を複数のハードウエアにインストールして使う場合は、ハードウエアの数だけ異なる証書シリアル番号(=ライセンス証書)が必要です。
- ライセンス認証済みのコンピュータで上記 a.～ d.までの何れかのコンポーネントを交換、あるいは増設する際にはハッシュ値が変わるので交換前に「ライセンス認証登録の抹消」を実施して頂きますようご注意願います。ハッシュ値の変更を検出すると自動的にライセンス認証を再び実施するように促すダイアログボックスが表示され、処理が中断されます。
- ライセンス認証に関する詳しい操作手順は「9.ライセンス認証の手順(52 頁)」にある記事を参照してください。

6.5. UnLoad 機能、及び Query 機能

- 1つの表、あるいはパーティションから取り出せる最大行数は約 42 億 9 千万(符号無し 32 ビット整数の最大値)行です。この制限を越える行が戻されると異常終了します。

- UnLoad 機能では表データのフルセットが出力されます。つまり、結合文や where/orderby/group by 句の指定など任意のクエリを使ってデータを出力することが出来ません。またシノニム、ビュー、スナップショット及び分散データベースのリモート表への問い合わせの結果を出力出来ません。このような用途には Query 機能をお使い下さい。
- ファイルサイズの上限については Windows API のファイルポインタとして符号無し 64 ビット整数を使っておりますので、内部的には 16EB（エクサバイト）が理論上の上限となりますが、実際に可能な最大サイズはファイルシステムの実装上の制約を受けます。マイクロソフト社が公表している資料によれば上限は次の通りです。
  Windows Server 2008 は 16TB まで
  Windows Server 2012 は 255TB まで
- BFILE ロケータの指し示す OS ファイルの内容は出力できません。
- 10g 未満の Oracle Client を御使用の場合、BLOB、CLOB、NCLOB の各データ型で出力可能なオブジェクトのサイズは 4GB-1・バイト/文字です。
- XMLType 型を含む表のアンロードへは対応しておりません。但し XMLType 型のタイプ・メソッド getClobVal() で修飾した問合せを Query 機能を使って実行することにより CLOB 型として XML データを出力することができます。CLOB 型として出力されたデータは SQL*Loader を使って XMLType 型へロードすることができます。getClobVal() の使用例についてはサンプル sample¥GetClobVal.sql をご覧ください。
- LONG, LONG RAW の各データ型の列を含む表の SQL*Loader 用制御ファイルは無条件にこれらの列をフィールドリストの末尾へ再配置します。これはこれらの列を含んでいる表でもダイレクト・パス・ロードが選択できるようにするための処置です。
- 主キー制約の施行に基づいて施行される NotNull 制約については有効性を切り替える為の ALTER TABLE{ENABLE/DISABLE} CONSTRANT …コマンドをスクリプト中に出力しません。
- 自動的に作成される制御ファイルは Oracle8i 以降の SQL*Loader に適合します。Oracle8 以下の SQL*Loader には対応していない構文を出力することがある為、使用できない場合があります。特に LOB 型データが含まれているファイルを Oracle7/8 の SQL*Loader を使ってロードするとエラーが発生します。
- 管理作業用のスクリプトを実行すると索引は常に nologging モードで作成されます。従ってロード終了後最初のバックアップを取得する前に媒体障害が発生するとこれらの索引を回復することが出来ません。ロードが終わりましたら速やかにデータ・ファイルのバックアップを取得して下さい。

  詳細は「16.. ダイレクト・パス・ロード（132 頁）」をご覧下さい。
- BFILE 型を出力するとき、ディレクトリ別名が指し示すディレクトリにファイルが実在するかどうかが検査されます。実在しない場合にはディレクトリ別名、ファイル名には文字列が出力されません。BFILE 型を登録したつもりなのに文字列が出力されない場合には、ディレクトリ別名の指し示している文字列が（アルファベットの大文字／小文字の区別も含めて）実在のディレクトリと一致するかどうかを確認して下さい。
- 再ロード後には表と索引のオプティマイザ統計が無効になるか、存在しても実態を正しく反映していない可能性があります。従いましてコストベースの最適化アプローチを使ってチューニングされているアプリケーションでは多くの場合パフォーマンスの劣化を引き起こします。このような現象が生じた場合にはオプティマイザ統計を収集しなおしてください。(ビルトインパッケージの DBMS_STATS を使います)
- UnLoad、Query 両機能の動作中は以下の2つの理由により他の接続によるトランザクションの実行をなるべく控えてください。

  a. データ出力中は文レベル（またはトランザクションレベル）の読取一貫性を保証するため、UNDO セグメント上の読取一貫性スナップショットと呼ばれる領域から更新前の行データ読み込まれます。この為、他の接続によって commit を伴う大規模なトランザクションが実行されると、これらのトランザクションによって生ずる UNDO によってこのスナップショットが上書きされ下記のエラーが発生し易くなるためです。

```
ORA-01555: スナップショットが古すぎます(ロールバック・セグメント番号 xx、名前 xx が小さすぎます)。
```

  b. 特に環境変数 CONSISTENT（76 頁）が指定されていない場合にはトランザクションの分離レベルが「Read Commited」となります。この分離レベルでは他のトランザクションによるコミットされた結果を何回

でも読み込んでしまう妨害 (Non-Repeatable Read と Phantom)を斥ける事が出来ません。その結果、参照整合性を満たさない行が出力されてしまうためです。

- SQL*Loader の仕様によりクラスタ表に対するダイレクト・パス・ロードは出来ません。またクラスタ表に対する TRUNCATE TABLE 文は無効であるため、制御ファイルへ TRUNCATE を指定したロードは出来ません。
- リファレンス・パーティション（参照パーティション）によって参照される、被参照パーティション表の Unload / Query は問題なく行えます。しかし、被参照パーティション表単独の再ロードは行わないで下さい。

  リファレンス・パーティション表の参照整合性制約はその定義により無効化できないため、SQL*Loder を使っての再ロード時に数々の制約を受けます。MTU はこの制約を十分クリアするための制御を行っていないので、間違って再ロードしてしまうと回復が難しいデータ消失を惹き起こす可能性があります。
  被参照パーティション表を再ロードする計画がある場合は必ず、参照パーティション表のデータもアンロードしてください。そして下記の順番で再ロードを実施してください。

  a. 参照パーティション表のデータ削除 (truncate table)
  b. 被参照パーティション表のデータ挿入 (REPLACE メソッドを使った SQL*Loader 実行)
  c. 参照パーティション表のデータ挿入(REPLACE or TRUNCATE メソッドを使った SQL*Loader 実行)

6.6. 固定長形式に含まれる空白文字

- 固定長形式で CHAR 型 NULL 可能のデータを再ロードする時には、半角空白文字列の二義性にご注意下さい。

  このようなデータ型の中にデータとしての半角空白文字列と NULL とが混在している場合に固定長形式を選択するのは（両者を区別できないので）適切ではありません。可変長データ形式を選択してください。
  半角空白文字列しかない場合は"PRESERVE BLANKS" オプション、NULL しかない場合は制御ファイルを編集し"NO PRESERVE BLANKS"（＝省略時初期値）と書換えてください。
- 上記と似ている理由ですが、固定長形式で VARCHAR2 型のデータを再ロードする時は半角空白文字列の多義性にご注意下さい。

  VARCHAR2 型を含む制御ファイルでは"PRESERVE BLANKS" オプションが書かれていないので、フィールド右側の半角空白文字は全て切り捨てられます。また、フィールドが全て半角空白文字の場合は、NULL として扱われます。
  経験的にこれが期待通りの動作であるケースが殆どですが、例えば

"右側にある空白３文字もデータの一部です___"　　（アンダースコア部分は半角空白とする）

というデータがあった場合、アンダースコア部分はストリップされ、もはやオリジナルでは無くなります。
また、制御ファイルへ"PRESERVE BLANKS"オプションを加えると、最大長分の空白文字列データとしてリロードする事ができますが、もともとのデータが何バイト分の空白文字列だったかという情報は失われてしまいます。
上記のような振る舞いが好ましくない場合は、やはり可変長データ形式を選択してください。

6.7. TIMESTAMP WITH LOCAL TIME ZONE 型の再ロード

SQL*Loader には TIMESTAMP WITH LOCAL TIME ZONE 型を次の条件が全て揃った状態でで再ロードするときに間違ったタイムゾーンでデータがロードされる障害があるのでご注意下さい。

現象的にはタイムゾーン±0のローカル時間でデータを解釈したような結果になります。

1. ダイレクト・パス・ロードを使用（従来パスでは発生しない）
2. リモートコンピュータからデータを再ロード（ローカルでは発生しない）
3. TIMESTAMP WITH LOCAL TIME ZONE 型の列のフィールド指定で下記を指定

```
＜列名＞　TIMESTAMP WITH LOCAL TIME ZONE ″＜書式マスク＞″ ENCLOSED BY '＜囲み文字＞'
```

この障害を回避するには、3つの条件がどれかひとつでも不成立となるように手作業でスクリプトを修正するかデータを移動してください。3を不成立とするには、下記の様に修正（赤字で示した部分を除去）してください。

```
＜列名＞　TIMESTAMP ″＜書式マスク＞″ ENCLOSED BY '＜囲み文字＞'
```

しかし、3の修正は1，2の条件のいずれかも不成立にした場合に逆に間違ったタイムゾーンでデータをロードしてしまう副作用もあるのでご注意下さい。

6.8. GetMeta 機能

- アドバンスト・レプリケーションには対応していません。
- CREATE SEQUENCE 文で指定される START WITH オプションの値は、ALL_SEQUENCES ビューの LAST_NUMBER の列値に由来します。従って次回 NEXTVAL によって取得できるよりも大きな値が戻されることが有ります。どのような値が戻されるかは CREATE SEQUENCE 文の CACHE オプションの設定値によります。

6.9. MTU.xla アドイン

- Excel が格納できる列数、あるいは行数の上限を超えて、問合せが結果を戻す場合、上限を超えた分のデータは切り捨てられます。上限数については、お使いの Excel のマニュアルなどでご確認下さい。

  また、問合せ結果のデータ量が非常に多い場合は、画面がハングしたような状態が数分間続く事がありますのでご留意下さい。ハングは待ち続ける事により解消します。

7. アップグレード

この章では旧版(v4.2未満)を運用中のお客様向けに製品のアップグレード方法を説明します。新規ご利用のお客様はこの章を読み飛ばして次の章から読み進める事が可能です。

7.1. v4.1 から v4.2へのアップグレード

v4.1 がインストールされている状態で v4.2のインストールパッケージをインストールすると、自動的に v4.1 はアンインストールされます。

旧版のアンインストールの過程で、もともとパッケージに含まれていたファイルは全て削除されますので、Env.bat などアップグレード後に引き継ぎたいものがありましたら事前にバックアップを取得しておいてください。

但し旧版を使用していた時に作成された製品フォルダ上のファイルのうち、インストール後に作成されたファイルは削除されずに残ります。製品フォルダの場所は:「表 11 製品フォルダとプログラムフォルダ(43 頁)」でご確認ください。

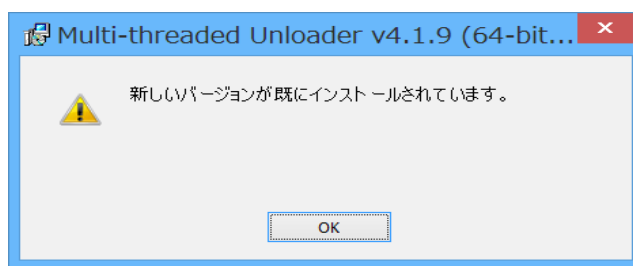
v4.1 を使用していた時にライセンス認証を行っていた場合は、旧製品フォルダ上に license.dat が作成されています。

エクスプローラーなどを使って新製品フォルダの同じ位置へコピーすると、そのまま認証データも引き継ぐことが出来ます[9]。

旧製品フォルダは不要であれば削除可能です。

7.2. v4.2から v4.1 へのダウングレード

旧版へダウンロードする場合、v4.2を一旦アンインストールしてから行ってください。

v4.2をアンインストールせずにいきなり旧版をインストールしようとすると右記のメッセージボックスが表示されて作業が中断されます。

7.3. v3.06 から v4.2へのアップグレード

7.3.1. v4.2の新機能を使用する場合

このバージョンから追加された新機能(12 頁)を使いたいお客様向けのアップグレード手順を下記に説明いたします。

- 今まで御使用になっていた旧版のインストール媒体をご用意下さい。

[9] 但し、機種変更、CPU 数の割増／換装、ディスク構成やネットワーク・コントローラ構成の変更がない場合に限ります。

- 旧版の製品ホームディレクトリのうち、お客様がカスタマイズして使っているファイルやディレクトリをリストア可能な媒体へ保存します。env.bat,signlcs.dat,license.dat と input, qryvar, qryfix の各ディレクトリを保存対象としてください。
- 「8..インストール (39頁)」にある手順に沿ってこのバージョンをインストールしてください。
- このバージョンの env.bat と保存してあった旧版の env.bat をエディタやマージ・ツールなどで開き、双方の版で共通する環境変数について、旧版の値でこのバージョンの値を書き換えてください。
- input, qryvar, qryfix の各ディレクトリの内容を保存してあった旧版の内容で置き換えてください。
- 旧版を使っていたときと同じ方法で製品を起動して、旧版と同じように動作するかどうかテストしてください。
- 上記テストで問題がなければアップグレードは完了です。
- 上記のテストで問題が発生した場合、画面出力結果と env.bat を添えて弊社サポート係へご相談下さい（但しサービス提供は保守サポート契約を予め締結されているお客様に限らせて頂いております）。

  このバージョンの製品ホームを残した上で、旧版を再度インストールし、セーブしてあった旧版の env.bat,signlcs.dat,license.dat や input, qryvar, qryfix の各ディレクトリを元に戻して原状回復してください。

7.3.2. v4.2の新機能を使用しない場合

このバージョンから追加された新機能（12 頁）は必要ないが、このバージョンで修正されたバグには対応したい、というお客様向けのアップグレード手順を下記に説明いたします。

このバージョンは旧版に対する上位互換性がありますので、原則的に製品の実行形式ファイルルを置換えて、いくつかのファイルをコピーするだけで旧版と同様に使用することが可能です。

- 旧版の MTU を利用している OS アカウントでログインします。
- x86 版のインストーラーを起動して新しい版の製品をインストールした後、新しい版の製品ホームディレクトリから次のファイルをを旧製品のフォルダへコピーして下さい。

  mtu.exe, PsPas.cer, PsPasCli.dll, PsPasCom.dll
- env.bat に記述された環境変数 EXECUTABLE の値を mtu_x86.exe から mtu.exe へ変更した後保存して下さい。
- 旧版を使っていたときと同じ方法で製品を起動して、旧版と同じように動作するかどうかテストしてください。ライセンス認証が完了していない場合はライセンス認証ウイザードが表示されるのでライセンス認証を完了させてからご利用ください（9.ライセンス認証の手順（52 頁）参照）。
- 上記テストで問題がなければアップグレードは完了です。
- 上記のテストで問題が発生した場合、画面出力結果と env.bat を添えて弊社サポート係へご相談下さい（但しサービス提供は保守サポート契約を予め締結されているお客様に限らせて頂いております）。

  前述の手順で変更した環境変数 EXECUTABLE の値を mtu.exe から mtu_x86.exe へ戻した後、env.bat を保存すると原状回復が完了します。

8. インストール

v3.06 以降から Query しか利用しない場合に限り、製品コンポーネント（MTU が接続先 DB で使用する4つの表と1つのストアドパッケージ）が無くても実行可能になり、これに関連するインストール作業が不要になりました。この場合は、

- 「8.1..旧版(v3.06)をお使いの場合 (39頁)」～「8.3..環境変数の編集と出力先ディレクトリ (45頁)」
- 「8.7..機能処理対象ユーザの指定(46頁)」～「8.8..接続先の指定(47頁)」

のインストール作業だけを実施してください。

Unload や GetMeta などの Query 以外の機能を使う場合には（オプション）と表示した以外の手順を全て実施する必要があります。

まとめると「表 9 利用機能別インストール手順（39 頁）」の通りです。

表 9 利用機能別インストール手順

インストール手順	頁	Query だけ使う	Query 以外も使う
8.1..旧版(v3.06)をお使いの場合	39	○	○
8.2..製品パッケージのインストール	40	○	○
8.3..環境変数の編集と出力先ディレクトリ	45	○	○
8.4..所有ユーザの準備	45	不要	○
8.5..システム権限の付与	45	不要	○
8.6..クオータの割当	46	不要	○
8.7..機能処理対象ユーザの指定	46	○	○
8.8..接続先の指定	47	○	○
8.9..EXCEPTION 表作成（オプション）	47	不要	任意
8.10..製品コンポーネントのインストール	48	不要	○

製品のインストールは Administrators グループに所属するユーザで実施して下さい。インストール終了後は通常の（Users グループ所属の）ユーザで製品を使用することが出来ます。

8.1. 旧版(v3.06)をお使いの場合

旧版の「マルチ・スレッド・アンローダー」をアンインストールされる場合にはその版の取扱説明書（readme.pdf）[10]に示されているアンインストール方法に従って完全にアンインストールしてください。

[10] v3.03 より旧い版では readme.txt をご覧下さい。

8.2. 製品パッケージのインストール

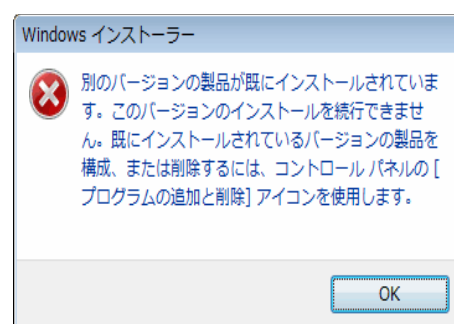
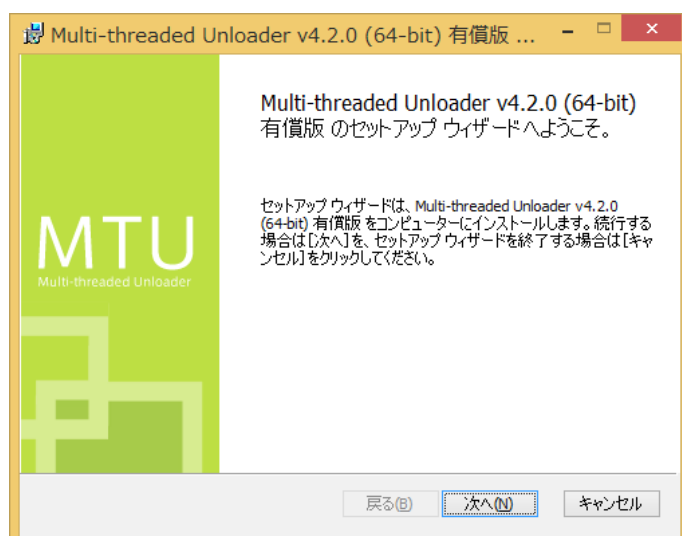
製品パッケージのインストール前に次の留意事項をご確認ください。

- インストール対象のコンピュータが「6.1.必要システム構成（31 頁）」に書かれた要件を満たしていることご確認ください。
- 同一製品の新しいバージョンと従来のバージョンは同じコンピュータに共存させる事はできません。
- 同一製品、同一バージョン、同一 bit 数のパッケージは同じコンピュータに1共存させる事はできません。このケースは有償版とサンプル版を共存させようとした場合に発生します。
- 同一製品の同一バージョンの x86 版と x64 版は同じ 64-bit ベースの Windows コンピュータに共存させる事が可能です。
- 同一製品の異なるパッケージを複数インストールしてある場合、後からそのいずれかをアンインストールしても他のインストール済み製品に影響は及びません。
- 本製品はインストール後の再起動を求められることなくインストールすることができます。但し、ジョブ管理用のミドルウエアを使用して製品をお使いになる場合に限り、インストール中に表 11:製品フォルダとプログラムフォルダ（43 頁）に記載のシステム環境変数を追加しますので、追加された環境変数を認識させるため、インストール後にエージェントの Windows サービスを再起動してくださいますようお願いします。

製品パッケージの含まれている CD-R 等の媒体をインストール対象のコンピュータで読み取り可能な状態にしてください。製品パッケージファイルには bit 数に応じて次の2種類があります。

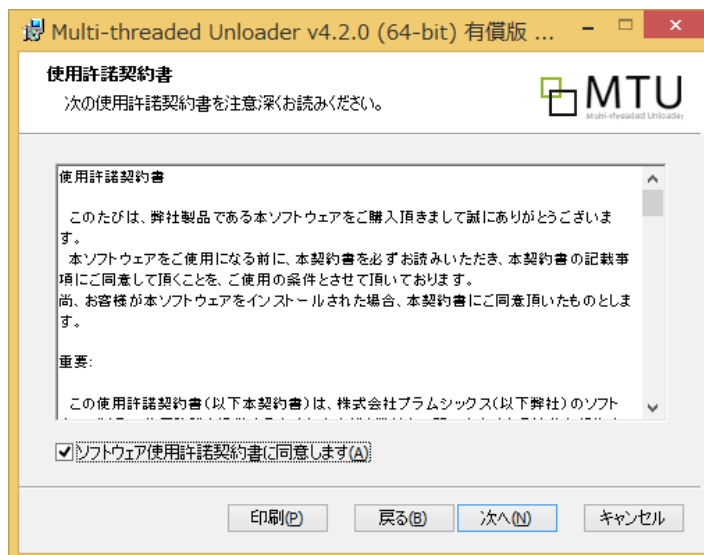
表 10: 製品パッケージと bit 数

製品パッケージファイル名	bit 数
mtu_x86.msi	32
mtu_amd64.msi	64

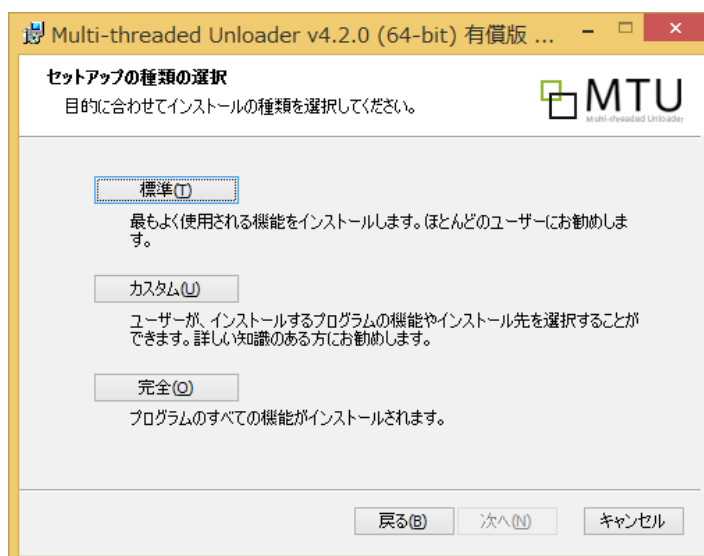
エクスプローラを起動して媒体中にある MSI 形式の製品パッケージファイルをダブルクリックします。次の左右いずれかの画面が表示されます。

左側の画面の場合は次へボタンを押してインストールを続けることができます。右側の画面の場合は同一 bit 数の有償版とサンプル版を同じコンピュータ上にインストールしようとしているケースで発生するエラーですので先にインストールされているものをアンインストールしてください。

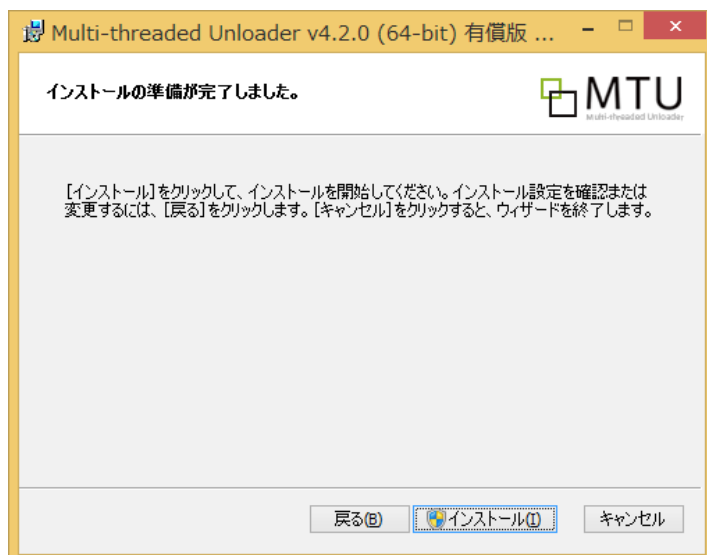
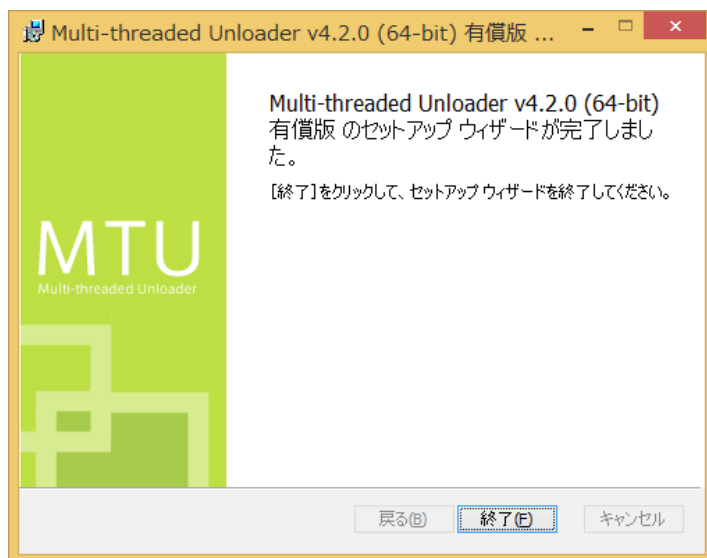
これから説明の中で使用する画面イメージはすべて 32-bit 版のものです。64-bit 版の画面構成は 32-bit 版と同じですので 64-bit 版のインストールでは 32 と表現されている箇所を 64、または x86 と表現されている箇所を x64 のように読み替えることが出来ます。

「次へ」を選択します。使用許諾契約書が表示されますので、内容をご確認ください。

ご同意いただけましたら「使用許諾契約に同意します」をチェックして「次へ」ボタンを押してください。同意されない場合には「キャンセル」ボタンを押して中止することができます。

次にセットアップの種類の選択が表示されます。

インストール場所やインストールするオプショナル・コンポーネントを取捨選択したい場合には「カスタム」ボタンを押します。特にカスタマイズに関するご要望がない限り「標準」を選んでください。

「完全」は現在のところ「標準」と同じインストール結果になります。

インストール開始してよろしければ「インストール」ボタンを押してください。

次の画面表示に変わるまで待機してください。環境によっては少し時間の掛かる事があります。

「完了」ボタンを押してインストールウイザードを終了させてください。

製品のインストールが完了すると、コントロールパネルの「プログラムと機能」へ製品タイトルが表示されます。

インストールが完了すると下記の場所に製品プログラムとドキュメントがインストールされます。

```
%ProgramFiles%¥PLUMSIX¥mtu_x.xx
```

（x.xx はバージョン符号、WOW64 を使って 32-bit 版をインストールした場合は%ProgramFiles%の部分を%ProgramFiles(x86)%）に読み替えて下さい）

説明の便宜上このフォルダをプログラムフォルダと呼ぶことにします。

また、下記の場所に製品用のシェルスクリプトと作業ディレクトリなどがインストールされます。

```
%ALLUSERSPROFILE%¥PLUMSIX¥mtu_x.xx¥yyy
```

（x.xx はバージョン符号、yyy は x86 か x64 の何れか）

説明の便宜上このフォルダを製品フォルダと呼ぶことにします。

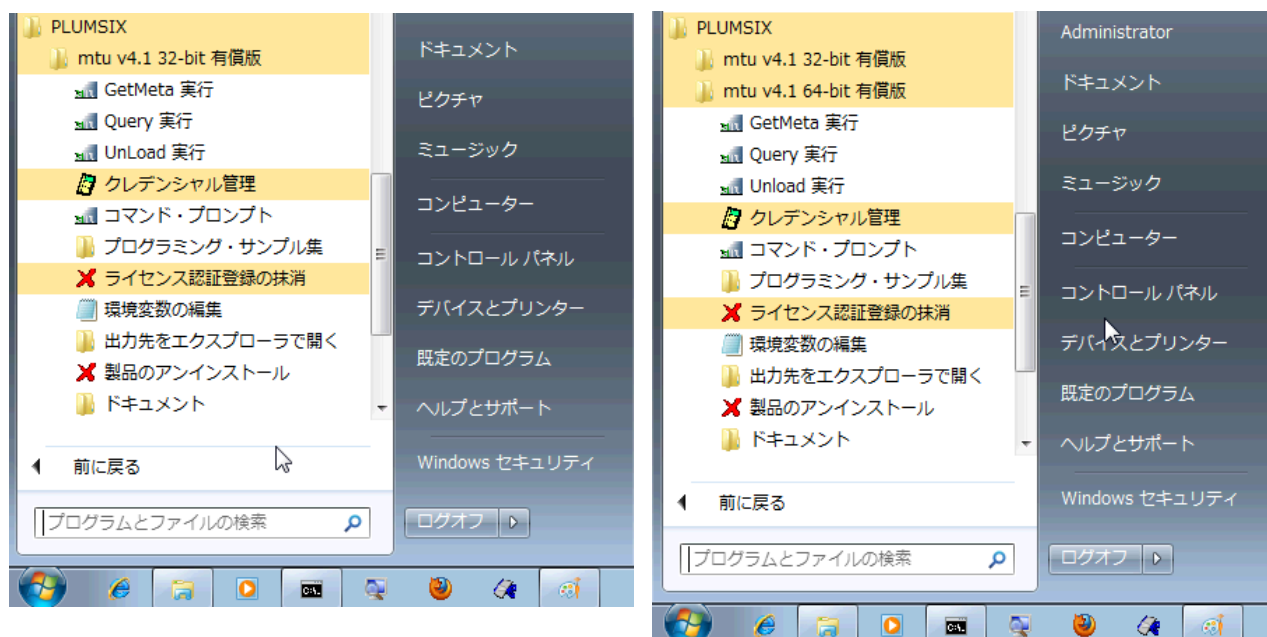
インストーラが完了するとプログラムフォルダを指し示す為のシステム環境変数として MTU_EXEC_DIR_X86（あるいは MTU_EXEC_DIR_X64）が、製品フォルダを指し示す為のシステム環境変数として MTU_USER_DIR_X86（あるいは MTU_USER_DIR_X64）が作成されます。

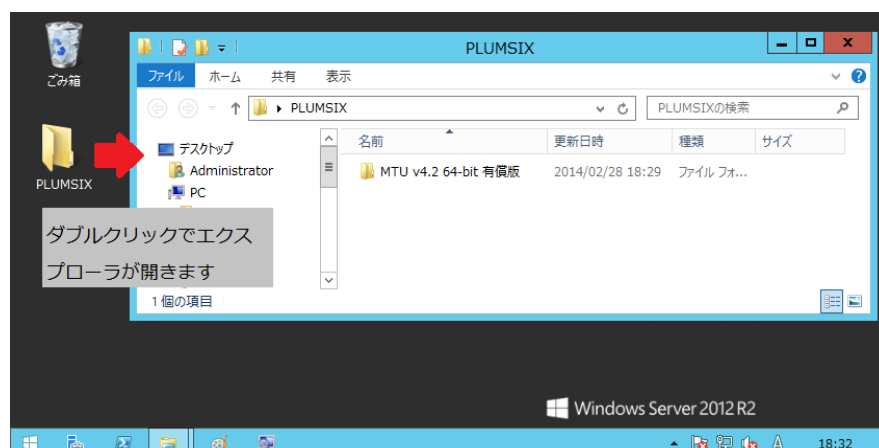
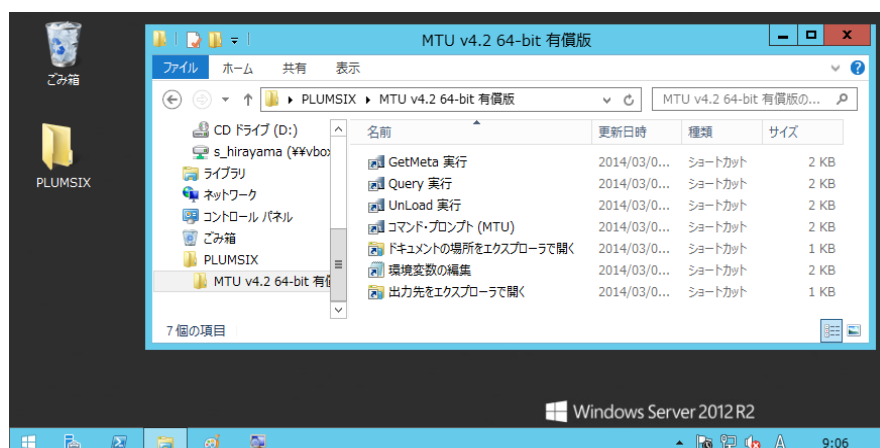
上記までの内容を下表にまとめます。

表 11:製品フォルダとプログラムフォルダ

本書内での呼名	パッケージ bit 数	システム環境変数名	設定される値
プログラムフォルダ	32	MTU_EXEC_DIR_X86	%ProgramFiles%¥PLUMSIX¥mtu_4.2¥x86 WOW64 の場合は %ProgramFiles(x86)%¥PLUMSIX¥mtu_4.2¥x86
	64	MTU_EXEC_DIR_X64	%ProgramFiles%¥PLUMSIX¥mtu_4.2¥x64
製品フォルダ	32	MTU_USER_DIR_X86	%ALLUSERSPROFILE%¥PLUMSIX¥mtu_4.2¥x86
	64	MTU_USER_DIR_X64	%ALLUSERSPROFILE%¥PLUMSIX¥mtu_4.2¥x64

製品フォルダは、これをオリジナルとしてコピーを別に作成しコピー上のスクリプト・ファイルを自由に編集して利用することも出来ます。詳しくは「31.製品フォルダのコピー方法（191 頁）」にある記事を参照して下さい。

Windows Vista/7/Server2008 では、インストーラが完了すると「スタートメニュー」->「すべてのプログラム」の下層に PLUMSIX フォルダが作成されます。下記にその画面イメージを示します。先に 32-bit 版、後に 64-bit 版を重ねてインストールした例です。

さらに、デスクトップ上に「PLUMSIX」フォルダが作成されます。

このフォルダをダブルクリックするとエクスプローラが起動され、MTU 用のショートカットを含むフォルダが現れます。

「MTU v4.2 xx-bit 有償版」と表示されたフォルダを開くと、ショートカットが現れます。

管理者以外のユーザ(グループ名:Users)に対して下表に示したフォルダへのアクセス権を与えられます。与えたア

クセス権はサブ・フォルダへ継承されます。下表に示したフォルダ以外の製品フォルダに対して、管理的なファイル操作を行うときは Administrators に所属するユーザでログインしてから実施して下さい。

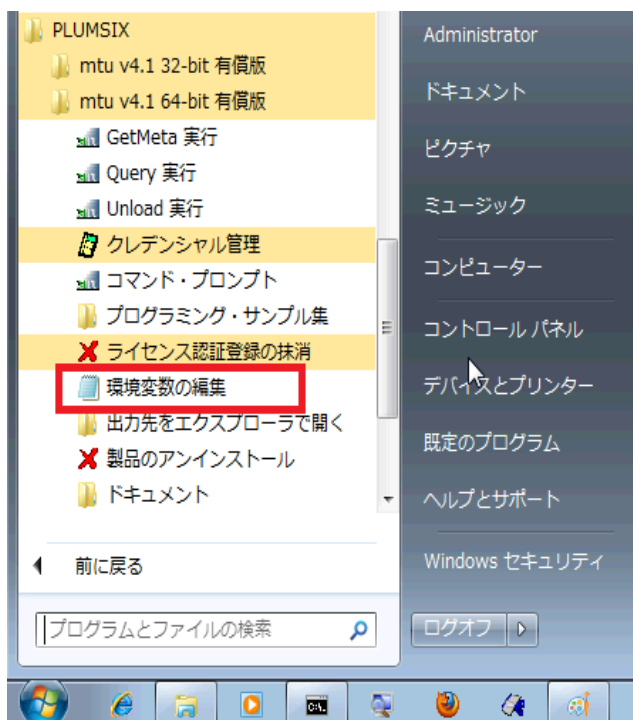
表 12:一般ユーザ (Users) に与えられたアクセス権

32-bit	64-bit	アクセス権
%MTU_USER_DIR_X86%	%MTU_USER_DIR_X64%	M (読取、書込、実行)

8.3. 環境変数の編集と出力先ディレクトリ

「スタート」→「すべてのプログラム」→「PLUMSX」→「mtu v4.2 ??-bit 有償版」→「環境変数の編集」

上記の製品メニューに含まれるショートカットを選択するか、製品フォルダ(43 頁)に含まれる Env.bat ファイルをテキストエディタで開きます。ショートカットの場所を右記に示します。

Env.bat ファイルを開いたら、環境変数 OUTPUT(89 頁)が指し示す値を必要に応じて編集してください。製品が出力するファイルをどこに格納させたいのかこの環境変数 OUTPUT へ設定する値で指し示す必要があります。

出荷時の設定値として %MTU_USER_DIR%input が設定されていますが、他のディレクトリへ結果を出力したい場合には、その出力先ディレクトリの名前を記述し保存します。また指定した値のディレクトリが存在しなければ mkdir コマンドを使って予め作成してください。

8.4. 所有ユーザの準備

製品コンポーネントを所有する為の Oracle ユーザ(以下所有ユーザと表記します)を用意します。この所有ユーザは既存のユーザでも新しく作成[11]してもどちらでも構いません。SYSTEM ユーザの使用を推奨します。

8.5. システム権限の付与

所有ユーザが製品コンポーネントを所有するために必要なシステム権限を付与されているかどうかを SQL*Plus な

[11] CREATE USER 文を使ってユーザを作ることが出来ます。

どを使ってそのユーザで接続した後、下記の SQL を両方実行して確認します。

```
select * from session_privs where privilege in ('CREATE PROCEDURE','CREATE ANY
PROCEDURE');
select * from session_privs where privilege in ('CREATE TABLE','CREATE ANY TABLE');
```

この問い合わせのどちらか一方でも1行も戻されない場合、権限が不足しているためインストール作業を継続することができません。インストール作業を中断し、DB 管理者へ連絡して権限を与えてもらって下さい。ロールを介したシステム権限の付与でも有効です。

非 DBA ユーザを所有ユーザとして使う場合には、追加で次の SQL を実行して権限を付与して下さい。SYSTEM ユーザであれば相当する権限を既にもっているので実施不要です。

```
GRANT SELECT ANY TABLE, SELECT_CATALOG_ROLE TO <所有ユーザ名>;
```

8.6. クオータの割当

所有ユーザが製品コンポーネントに含まれる表や索引を所有するために必要な領域割当を受けているかどうかを SQL*Plus などを使ってそのユーザで接続した後下記の SQL を両方実行して確認します。

```
select default_tablespace, temporary_tablespace from user_users;
select * from user_ts_quotas;
```

この問い合わせで戻された情報により SYSTEM 表領域以外のデフォルト表領域と一時表領域、及びデフォルト表領域に対するクオータが所有ユーザへ割り当てられているかどうかを確認します。

割り当てが無い場合には DB 管理者へ連絡して SYSTEM 表領域以外のデフォルト表領域と一時表領域、及びデフォルト表領域に対するクオータを与えてもらって下さい。

クオータを割当てない場合は UNLIMITED TABLESPACE システム権限の付与でも有効です。その場合の確認の SQL は下記の通りです。

```
select * from session_privs where privilege = 'UNLIMITED TABLESPACE';
```

8.7. 機能処理対象ユーザの指定

各機能処理の対象とするスキーマ名を Env.bat の環境変数 SRC_USER(97 頁)へ指定するように編集してください。Oracle 11g 以降のデフォルトではスキーマ名に含まれるアルファベットの大文字と小文字が区別されるので注意して下さい。また出荷時の設定では SCOTT が設定されていますがこのユーザが全てのデータベースで存在するとは限り

ません[12]。

8.8. 接続先の指定

環境変数設定バッチファイル Env.bat 内の環境変数 USERID へ接続先を指定してください。USERID は実行形式ファイル（mtu.exe）の第一引数として使用されています。引数の指定方法については「25.プログラムの引数（168 頁）」に示された記事を参照してください。

Query のみ使用する場合は、問い合わせ対象データのオーナーか、あるいは SELECT ANY TABLE システム権限を持つ Oracle ユーザを指定します。Query 以外も使う場合は所有ユーザ（「8.4..所有ユーザの準備 (45頁)」参照）に編集します。

例えば Oracle Net サービス名：oracle.world の SYSTEM ユーザが接続先である場合には、下記のように記述します。

```
set USERID=SYSTEM/MANAGER@oracle.world
```

外部ユーザ(identified externally オプション付きの create user 文で作成されたユーザ)も使用することができます。この場合は下記のように記述します。

```
set USERID=/
```

Oracle 10g から登場した簡易接続ネーミング・メソッド[13]を使用することも出来ます。簡易接続ネーミング・メソッドでは「[//]host[:port][/service_name]」と言う書式を指定して接続先のデータベース・サービスを識別します。ここで各カッコ[]内の文字列はオプションです。この場合は例として下記のように記述します。簡易接続ネーミング・メソッドを使用する際は競合を避けるため環境変数 TNS_ADMIN に値が設定されていないことをご確認ください。

```
set USERID=SYSTEM/MANAGER@//localhost:1521/oracle.world
```

v3.06 からサポートされたパスワード秘匿化機能を使用する場合には下記のように記述します。この機能に関する詳細は「15.パスワードの秘匿化について（128 頁）」にある記事を参照してください。

```
set USERID=@userid.crd
```

8.9. EXCEPTION 表作成（オプション）

4.5.2.整合性制約の無効化／有効化切替機能（18 頁）を使う場合は実施してください。DB 管理者へ連絡して静的データ・ディクショナリ・ビューの EXCEPTIONS 表（例外表）とシノニムを作成してもらってください。EXCEPTIONS が既に作成済であるかどうかを確認するには、下記の問い合わせを実行してください。

[12] SCOTT スキーマは?/rdbms/admin/utlsampl.sql スクリプトを実行することによって作成することが出来ます。

[13] Oracle 製品マニュアル Oracle Database Net Services 管理者ガイド 10g リリース 2（10.2）参照

```
select owner,substrb(object_name,1,30) as object_name, object_type, status
from all_objects
where object_name = 'EXCEPTIONS';
```

EXCEPTIONS 表のパブリックシノニムを作成しておくと各実行ユーザが共有できます。EXCEPTIONS 表とパブリックシノニムの作成手順の例を下記に示します。

```
connect <所有者ユーザ名>/<所有者パスワード>@接続文字列
@?/rdbms/admin/utlexcpt
grant select,insert,update,delete on exceptions to public;
create or replace public synonym exceptions for exceptions;
```

8.10. 製品コンポーネントのインストール

Env.bat ファイルの編集が終わったら保存してください。

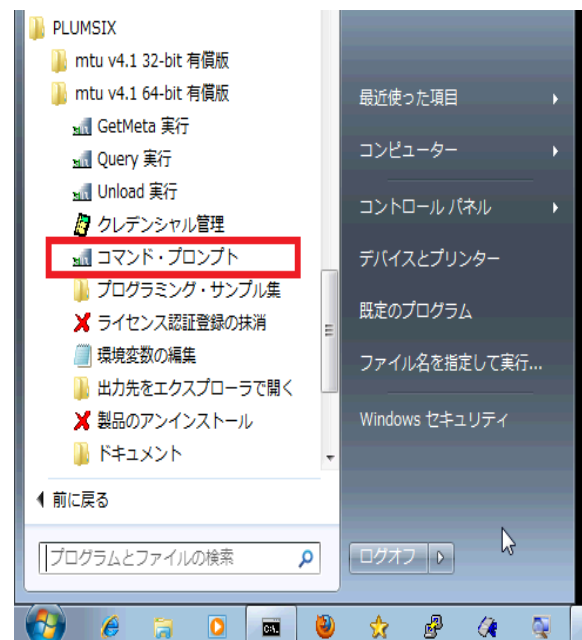
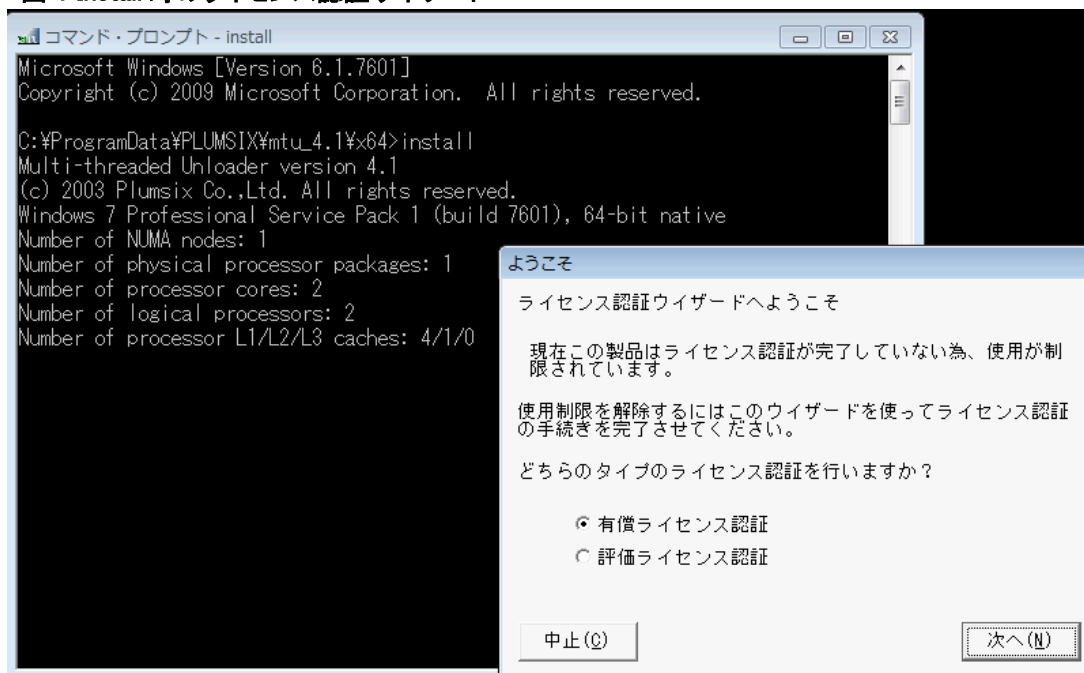
次に製品のメニューから「コマンドプロンプト」を選択してください。図は「コマンド・プロンプト」のある場所を示しています。

製品フォルダ(43 頁)にある、Install.bat シェル・スクリプトを実行してください。ライセンス認証が済んでいない場合、「ライセンス認証ウイザードへようこそ」というキャプションのダイアログボックス(下図)が表示されます。

図 1 Install 時のライセンス認証ウイザード

ライセンス認証の目的については「6.4.ライセンス認証の概要（33 頁）」にある記事をご確認ください。ライセンス認証を実際に行うには「9.ライセンス認証の手順（52 頁）」にある手順に沿って操作してください。

ライセンス認証済みの場合は Install.bat の残りの処理が実行されます。正しく実行された場合の画面の事例を下記に示します。

```
c:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64>install
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
Windows 7 Professional (build 7600), 64-bit native
Number of NUMA nodes: 1
Number of physical processor packages: 1
Number of processor cores: 2
Number of logical processors: 2
Number of processor L1/L2/L3 caches: 4/1/0
製品コンポーネントのインストールを実行します...
クライアント・キャラクタセット ID は 838 (JA16SJISTILDE) です。
ネット・サービス名:xxxxxxxxxxxxxx へユーザ名:SYSTEM として接続します。
しばらくお待ちください...
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options
接続先は SYSTEM@xxxxx.xxxxxxx (ネット・サービス名:xxxxxxxxxxxxxx) です。
サーバー・タイプは DEDICATED です。
監査接続ＩＤは 4013368 です。
サーバーの日付時刻は 2012-03-08 11:56:56 です。
対象スキーマは SCOTT です。
以下の製品コンポーネントを SYSTEM スキーマへインストールしました。
APLDDP01 (PACKAGE), APLDDT01 (TABLE), APLDDT02 (TABLE), APLDDT03 (TABLE), APLDDT04
(TABLE)
データベース接続用スクリプトを作成しました 。
```

```
c:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64¥input¥_connect.sql
アンインストール用スクリプトを作成しました 。
c:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64¥input¥_drp_prod.sql
パブリックシノニム作成用スクリプトを作成しました 。
c:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64¥input¥_cre_pub_syn.sql
パブリックシノニム削除用スクリプトを作成しました 。
c:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64¥input¥_drp_pub_syn.sql
処理対象となった表の総数      :            5
ラウンドトリップ（回）        :           28
出力バイト総数（kbytes）      :         2.51
所要時間（秒）                :         0.69
処理速度（kbytes/秒）         :         3.64
正常終了しました。

c:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64>
```

製品コンポーネントが既に作られている場合、install.bat は次のようなメッセージを出力して異常終了します。「ORA-00955: すでに使用されているオブジェクト名です。」というメッセージを含んでいる場合、これは無視することができます。

```
しばらくお待ちください...
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options
接続先は SYSTEM@xxxx.xxxxxxx（ネット・サービス名:xxxxxxxxxxxxxx）です。
サーバー・タイプは DEDICATED です。
監査接続ＩＤは 4013432 です。
サーバーの日付時刻は 2012-03-09 16:23:55 です。
対象スキーマは SCOTT です。
APLDDT01 表作成               :    Executing
ORA-00955: すでに使用されているオブジェクト名です。
関数 rcSqlStmtExecute() が異常終了しました。(c0ciStmt.c-448)
関数 rcExecDefineAndFetch() が異常終了しました。(c0ciStmt.c-1131)

Error offset: R=1, C=13
Line No. Statement
-------- 0----+----1----+----2----+----3----+----4----+----5
       1| create table APLDDT01 (
       2|   table_name varchar2(30)
       3|   , data_fmt number(2) default 0 not null
       4|   , constraint APLDDT01_PK primary key (table_name)
       5|   , constraint APLDDT01_CK01 check (
       6|     data_fmt in (0, 1)
       7|   )
       8| ) pctfree 0 storage (initial 128k next 128k pctincrease 100)

関数 rcOciStmtVa() が異常終了しました。(c0ciStmt.c-1505)
関数 rcOciStmtList() が異常終了しました。(c0ciStmt.c-1574)
関数 rcInstall() が異常終了しました。(rcInstall.c-567)
所要時間（秒）                :         0.25
異常終了しました。(mtu.c-487)
続行するには何かキーを押してください . . .
```

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64>
```

上記までの操作で環境変数 USERID に設定されたスキーマ内へ製品コンポーネントを構成する以下のデータベース・オブジェクトが作成されます。SQL*Plus の describe コマンドや user_objects 表などで確認して下さい。

表 13 製品コンポーネントに含まれる表とPL/SQL

名前	種類
APLDDT01	表
APLDDT02	表
APLDDT03	表
APLDDT04	表
APLDDP01	パッケージ仕様（本体は無し）

確認する為の問い合わせ文の例：

```
select substrb(object_name,1,30) as object_name, object_type, status
from user_objects
where object_name like 'APLDD___';
```

製品コンポーネントが作成されると引き続いてアンインストール時に使用する以下のファイルが環境変数 INPUT（83 頁）に示される位置へ作成されます。

```
_connect.sql
_drp_prod.sql
```

製品コンポーネントを構成するデータベース・オブジェクトを所有ユーザ以外の実行ユーザが共用できるようにする為のスクリプトが環境変数 INPUT に示される位置へ作成されます。

```
_cre_pub_syn.sql
_drp_pub_syn.sql
```

以上で終了です。

9. ライセンス認証の手順

この章ではライセンス認証の手順について解説します。ライセンス認証の概要については「6.4.ライセンス認証の概要(33 頁)」にある記事をご確認ください。

有償ライセンス認証の場合は手続きにあたり事前に発行された「ライセンス証書」が必要となりますので、お手元にご用意下さい。評価ライセンス認証の場合は手続きにライセンス証書は不要です。

ライセンス認証は Administrators グループに所属するユーザでログインした上で実施してください。

尚、ライセンス認証を開始する前に、Windows 更新プログラムが最新状態になっている事をご確認ください。ライセンス認証中に、更新プログラムの取得等が行われていると認証に失敗する事例が報告されている為です。

9.1. 認証登録

ライセンス認証を開始する為の特定のコマンドやショートカットは用意されていません。その代わり、MTU のプログラムの本体である mtu.exe が起動されるときに下記の場所にあるライセンスファイルをチェックします。

```
%ALLUSERSPROFILE%¥PLUMSIX¥mtu_4.2¥license.dat
```

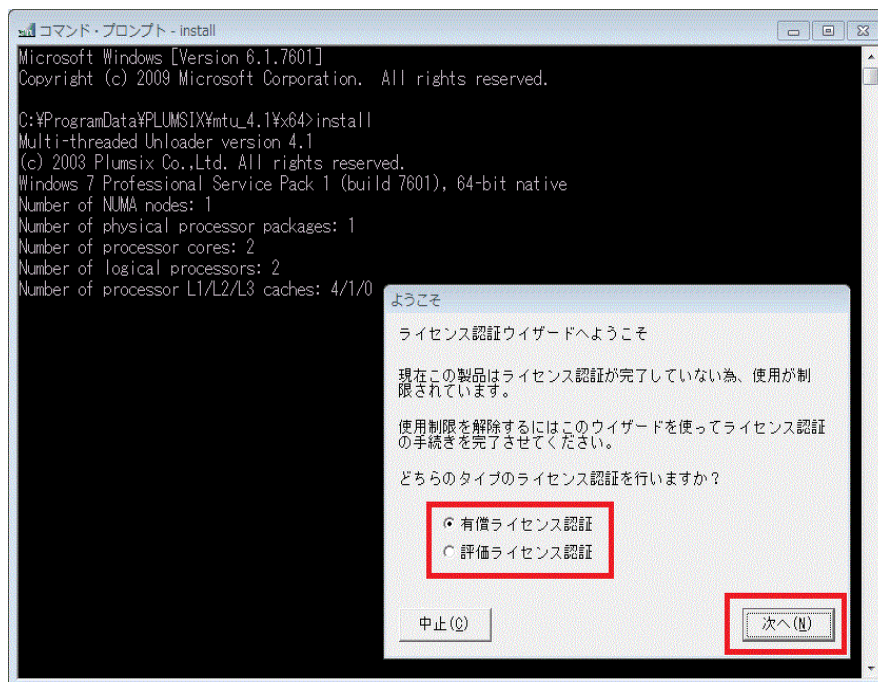
ライセンスファイル自体が存在しないか、ファイルに含まれるハードウエア情報と実際のハードウエア情報に差異を検出した時、または当日の日付が有効期間の範囲内にない場合にライセンス認証ウイザードが起動します。最初に表示されるのは「ようこそ」画面です。

9.1.1. ようこそ

「ようこそ」画面ではこれから登録しようしているライセンスが、有償かあるいは評価なのかを選ぶことができます。

お手元にライセンス証書がある場合、ラジオボタンで「有償ライセンス認証」を選び手続きを開始できます。

評価ライセンスで MTU を使用する場合はラジオボタンで「評価ライセンス認証」を選んでください。

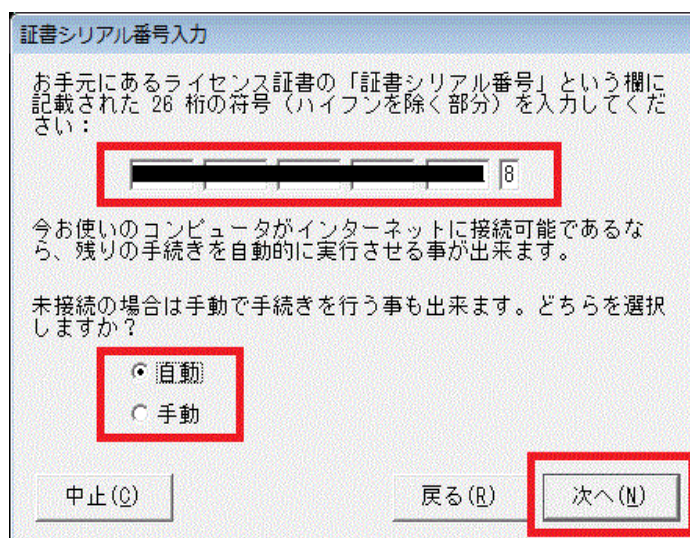
ラジオボタンを選択後、「次へ」をボタンを押してください。

「有償ライセンス認証」が選択されていた場合「証書シリアル番号入力」画面へ遷移します。次に「9.1.2.証書シリアル番号入力（53 頁）」に進んでください。

「評価ライセンス認証」が選択されていた場合「申請データ送信（手動）」画面へ遷移します。次に「9.1.4.申請データ送信（手動）（55 頁）」に進んでください。

9.1.2. 証書シリアル番号入力

「証書シリアル番号入力」画面では、6つ並んだエディットボックスへ手元にあるライセンス証書の「証書シリアル番号」を読み取って入力してください。左側から5つ目までのエディットボックスはそれぞれ5桁、もっとも右側のエディットボックスは1桁の数字かアルファベット大文字が入力可能です[14]。

ラジオボタンの「自動」と「手動」はそれぞれ、状況に応じて使い分けます。

「自動」はインターネットを使える場合に選択できます。この画面の後の操作が大幅に自動化されるのでライセンス認証を速やかに完了させることができます。

「手動」は下記いずれかの場合に当てはまるとき選択してください。下記の場合にはてはまらなくても、認証サーバ

[14] エディットボックスに入力された文字は規定の文字数を満たしていることと、最終桁（26桁目）のチェックディジットが他の25桁から求めたチェックディジットと一致するかがテストされるので、入力ミスや読み取りエラーが確実に排除されます。

ーやインターネットのトラブル発生時にこちらの利用をお願いすることもございます。

- 認証するコンピュータがインターネットへ繋がっていない時
- 評価ライセンス認証を行う時
- 「9.2.登録抹消（62 頁）」に説明した手順を経ずに同じ証書シリアル番号を使って別なコンピュータでライセンス認証をやり直す場合

＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊

手動ライセンス認証は係員による電話とメールを使用した窓口対応となりますので、迅速な対応を望まれる場合には、認証作業当日の数日前に遡り、「33.製品に関する問合せ先（190 頁）」の連絡先へ事前のご通告をお願いします。

＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊

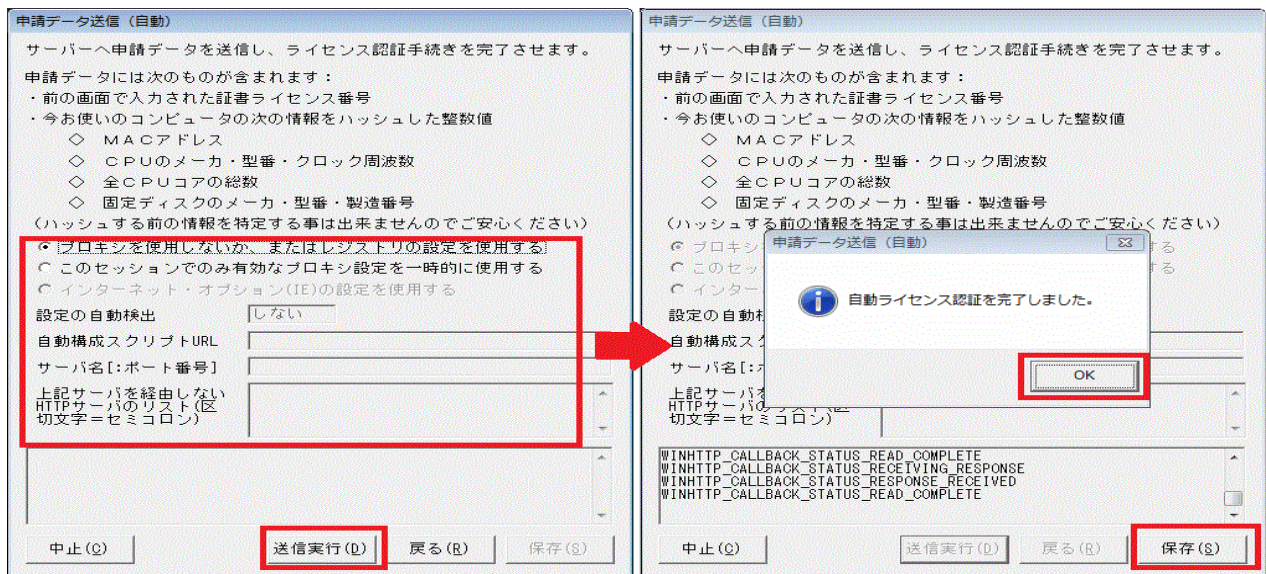
証書シリアル番号を入力し、ラジオボタンを選択後、「次へ」をボタンを押してください。

「自動」が選択されていた場合「申請データ送信（自動）」画面へ遷移します。次に「9.1.3.申請データ送信（自動）（54 頁）」に進んでください。

「手動」が選択されている場合「申請データ送信（手動）」画面へ遷移します。次に「9.1.4.申請データ送信（手動）（55 頁）」に進んでください。

9.1.3. 申請データ送信（自動）

「申請データ送信（自動）」画面では、前の画面で入力された証書シリアル番号とハードウエア情報を対にして「申請データ」としてまとめたものをライセンス認証サーバへ送信します。「送信実行」ボタンを押したタイミングで申請データが送信されます。

ライセンス認証の対象コンピュータが、直接インターネットへ接続することが認められておらず HTTP プロキシーを経由する利用のみ認められている場合は、3つあるラジオボタンの内2番目と3番目の選択を通じてプロキシー情報を構成することができます。設定手順の詳しい内容については「9.1.9.プロキシー情報の構成（61 頁）」に書かれた手順を参照してください。

行」ボタンを押すと次の操作が逐次実行されます。

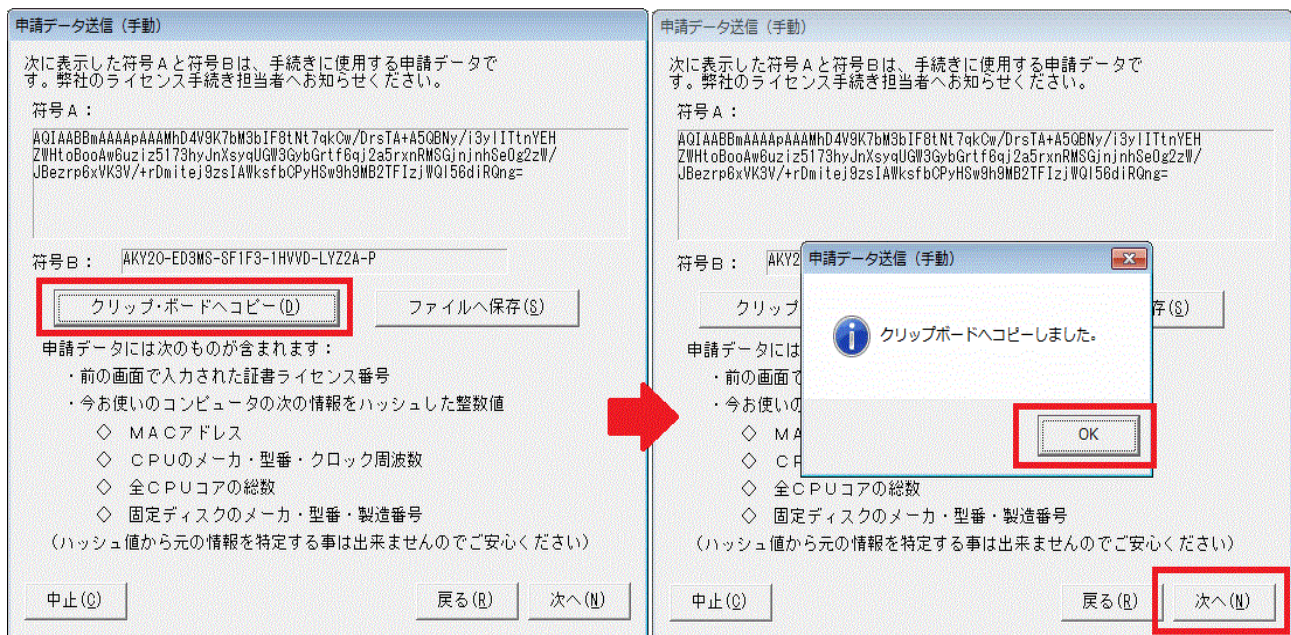
- 認証ウイザードが申請データの送信が開始するため、認証サーバーへ接続を試みます。
- 申請データが認証サーバーで認証されると、認証データが送られてきます。
- 認証データを受け取りライセンスデータの作成に成功すると「自動ライセンス認証を完了しました」ダイアログが表示されます。

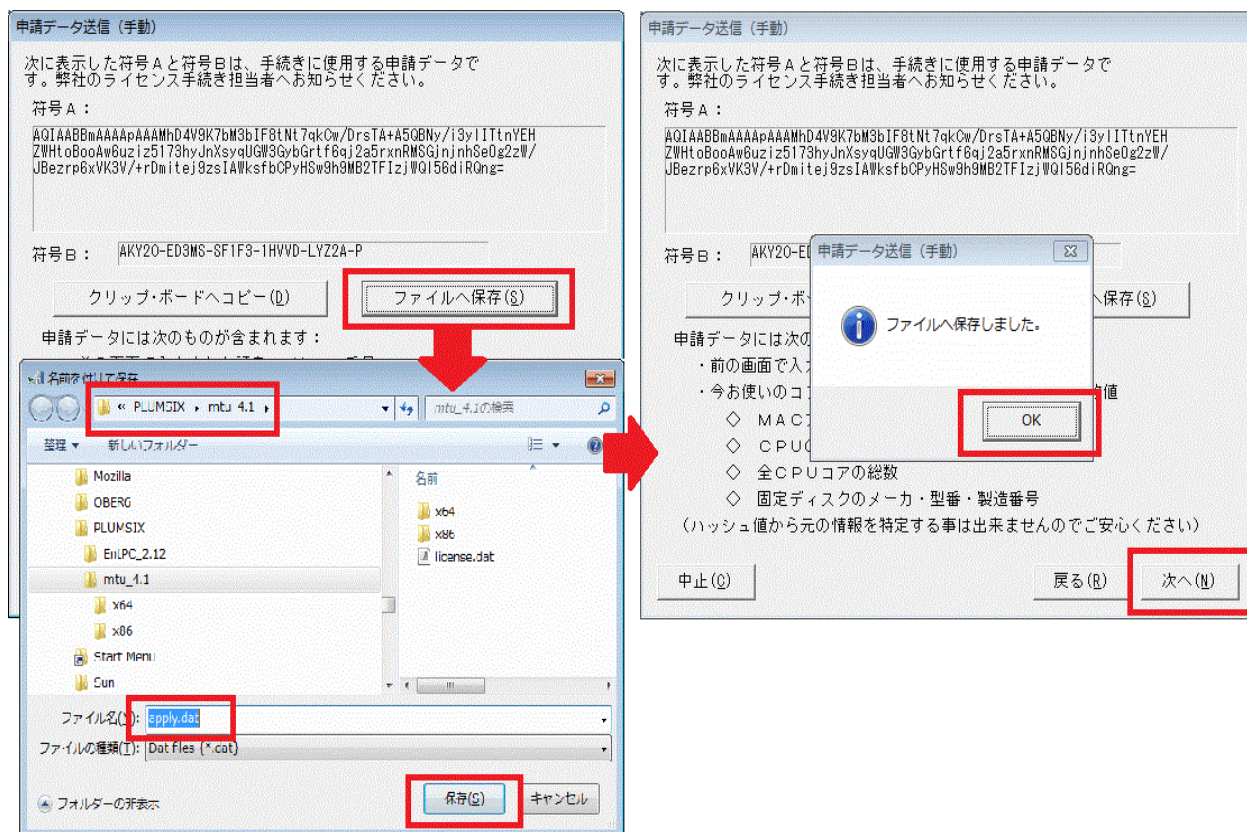
ダイアログを閉じると「申請データ送信（自動）」画面のボタンコントロールの幾つかの有効性が変更されます。ラジオボタンと「送信実行」と「戻る」ボタンが無効化され、「保存」ボタンが有効化されます。

ここで「保存」ボタンを押して下さい。次に「9.1.8.ライセンス認証結果の確認（60 頁）」に進んでください。

9.1.4. 申請データ送信（手動）

「申請データ送信（手動）」画面では、電子メールの本文や添付ファイルとして申請データを作成するための作業を行うことができます。「クリップ・ボードへコピー」か「ファイルへ保存する」のいずれか一つ、状況に適した手順を選択してください。

「クリップ・ボードへコピー」ボタンを押すと画面に表示されている「符号 A」と「符号 B」を元にした申請データが作成され、コンピュータのクリップボードへコピーされます。コピーが成功すると「クリップボードへコピーしました」というダイアログボックスが表示されます。

このダイアログを確認後は、電子メールの本文に CTRL+C（あるいはメニューバーから貼り付け）の操作をしていただくことにより、手動で申請データを送ることができます。申請データは既に暗号化された状態なのでお客様が申請データを暗号化する必要はございません。

「申請データ送信（手動）」画面で、「ファイルへ保存」ボタンを選ぶと,「名前を付けて保存」ダイアログボックスが表示されます。初期表示では製品フォルダを現在のディレクトリ、apply.dat という名前を採用しようとしますが、任意の場所、任意の名前に変更することも可能です。「保存」ボタンを押して保存に成功すると「ファイルへ保存しました」というキャプションのダイアログボックスが表示されます。

保存された申請データファイルをメールの添付ファイルとして送ることができます。申請データは既に暗号化された状態なのでお客様が申請データを暗号化する必要はございません。

申請データを電子メールで送信した後は「次へ」ボタンを押して次の「9.1.5.認証データ受取方法の確認(手動)（57頁）」に進んでください。

9.1.5. 認証データ受取方法の確認(手動)

＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊

申請データを送信してから認証データを受け取るまでの間、ライセンス認証ウイザードの画面は**開いたまま**にしておいて下さい。

画面を閉じてしまった場合は、途中からの**再開はできません**ので再び「9.1.1.ようこそ（52 頁）」からやり直して下さい。

＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊ 重要 ＊＊＊

「認証データ受取方法の確認（手動）」画面では、認証データの受取方法をについて状況に適したもの選ぶことができます。どのような場合にどちらを選ぶべきかのポイントがダイアログ上のキャプションに表示されていますので状況に適したものを選んでください。ラジオボタンを選択後、「次へ」ボタンを押してください。

「電話で受取る」ボタンが選択されていた場合、「認証データ受取（電話）」画面へ遷移します。続いて「9.1.6.認証データ受取（電話）（58 頁）」の手順に進んでください。

「メールで受取る」ボタンが選択されていた場合「認証データ受取（メール）」画面へ遷移します。続いて「9.1.7.認証データ受取（メール）（59 頁）」の手順に進んでください。

9.1.6. 認証データ受取（電話）

「認証データ受取（電話）」画面では、ライセンス認証担当者から電話で告げられた認証データを入力し、続いてライセンス証書に記載されたライセンス・キーを入力するための画面です。

ライセンス・キー入力ではライセンス証書に記載されたライセンス・キーを一つづつ「追加」ボタンを押してリストボックスへ追加してください。

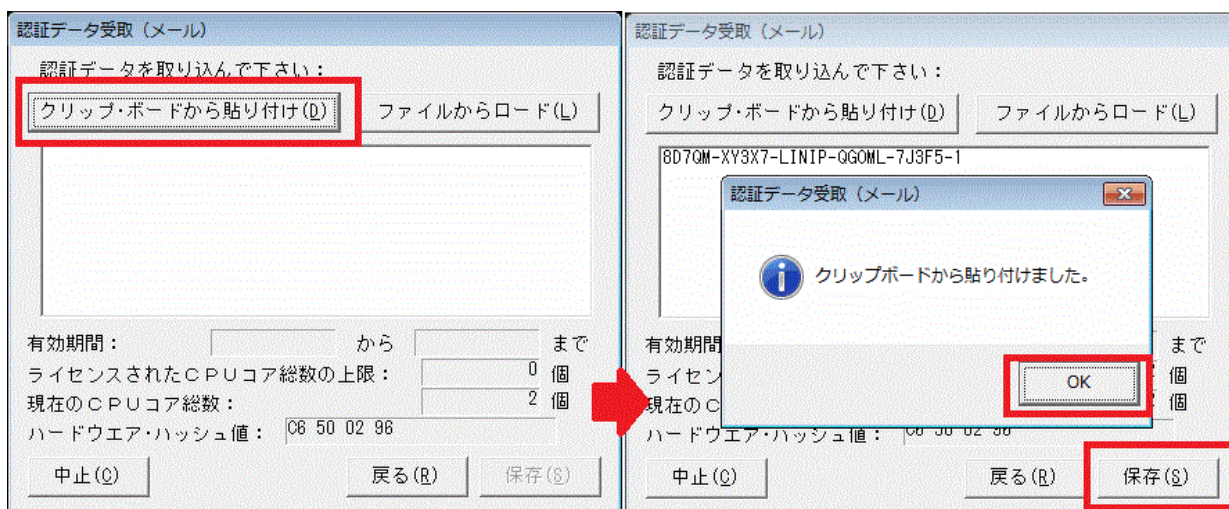
追加ボタンを押すごとにそのライセンスキーに含まれている CPU コア数の情報が加算され合計した数が「ライセンスされた CPU コア総数の上限」というフィールドへ表示されます。この数が「現在のCPUコア総数」というフィールドの

数以上にならないと製品を動作させることがで
きません。

認証データのフィールドを入力し、ライセンス証書に記載されたライセンス・キーをすべてリストボックスへ表示させることが可能になったら「保存」ボタンを押してください。次の「9.1.8.ライセンス認証結果の確認（60 頁）」画面の手順に進んでください。

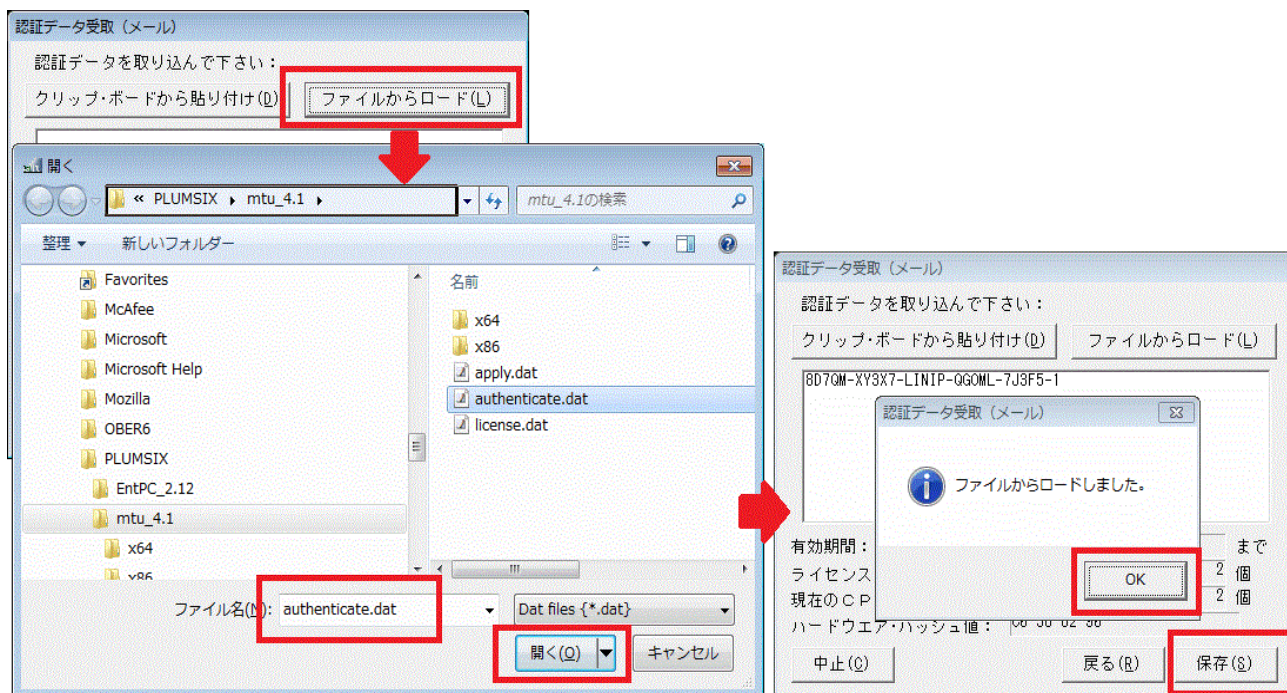
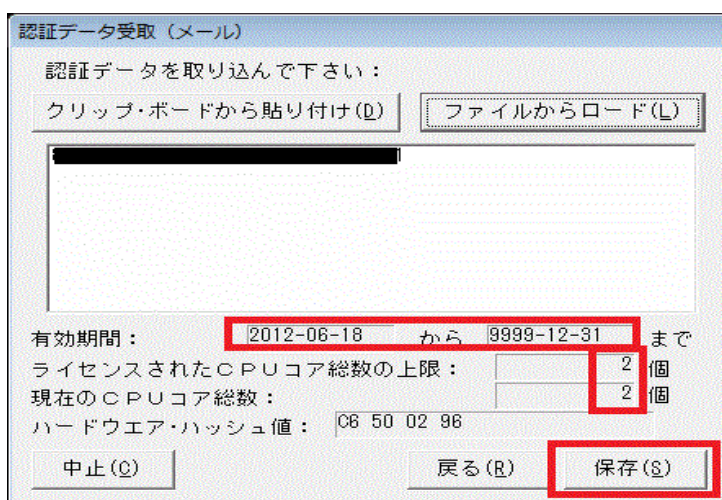
9.1.7. 認証データ受取（メール）

「認証データ受取（メール）」画面では、電子メールの本文や添付ファイルとしてライセンス認証担当者から送られてきた認証データを取り込むための作業を行うことができます。「クリップ・ボードから貼り付け」か「ファイルからロード」のいずれか一つ、状況に適した手順を選択してください。

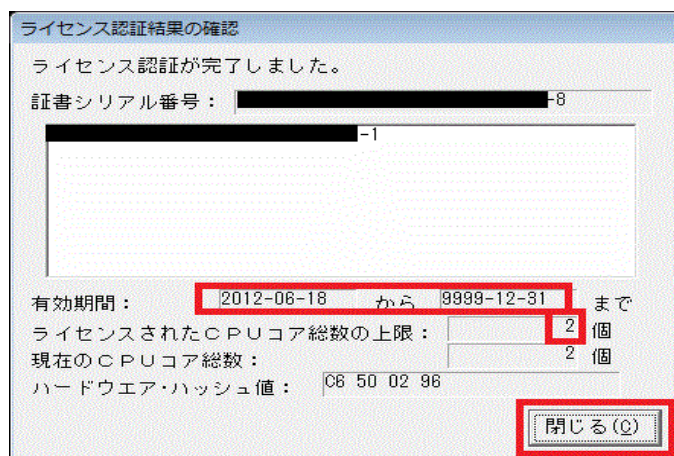
「クリップ・ボードから貼り付け」ボタンを押すと、現在クリップ・ボードに含まれている内容からライセンス認証ウイザードへ認証データを貼り付ることができます。従いましてこのボタンを押す前に事前にライセンス認証担当者から届いた電子メールの本文から認証データにあたる部分を選択してクリップ・ボードへコピーしておいて下さい。

認証データ以外のデータがクリップ・ボードに含まれている時の動作については未定義です。

「クリップ・ボードから貼り付け」に成功すると「クリップボードから貼り付けました」というダイアログボックスが表示されます。

このダイアログを確認後は、「認証データ受取（メール）」画面のコントロールが更新されています。

「ファイルからロード」ボタンを押すと、「開く」ダイアログボックスが表示されます。初期表示では製品フォルダを現在のディレクトリ、authenticate.dat という名前のファイルを探します。任意の場所、任意の名前に変更することも可能です。「開く」ボタンを押してオープンに成功すると「ファイルからロードした」というキャプションのダイアログボックスが表示されます。

「クリップ・ボードから貼り付け」および「ファイルからロード」ボタンの下のリストボックスに、認証データに含まれていた一つ以上のライセンス・キーが表示されます。その外に「有効期間」「ライセンスされたCPUコア総数の上限」「現在のCPUコア総数」の各フィールドが更新されます。保存ボタンを押して次の「9.1.8.ライセンス認証結果の確認（60 頁）」画面に進んでください。

9.1.8. ライセンス認証結果の確認

「ライセンス認証結果の確認」画面では、ライセンス認証処理結果のサマリーが表示されます。

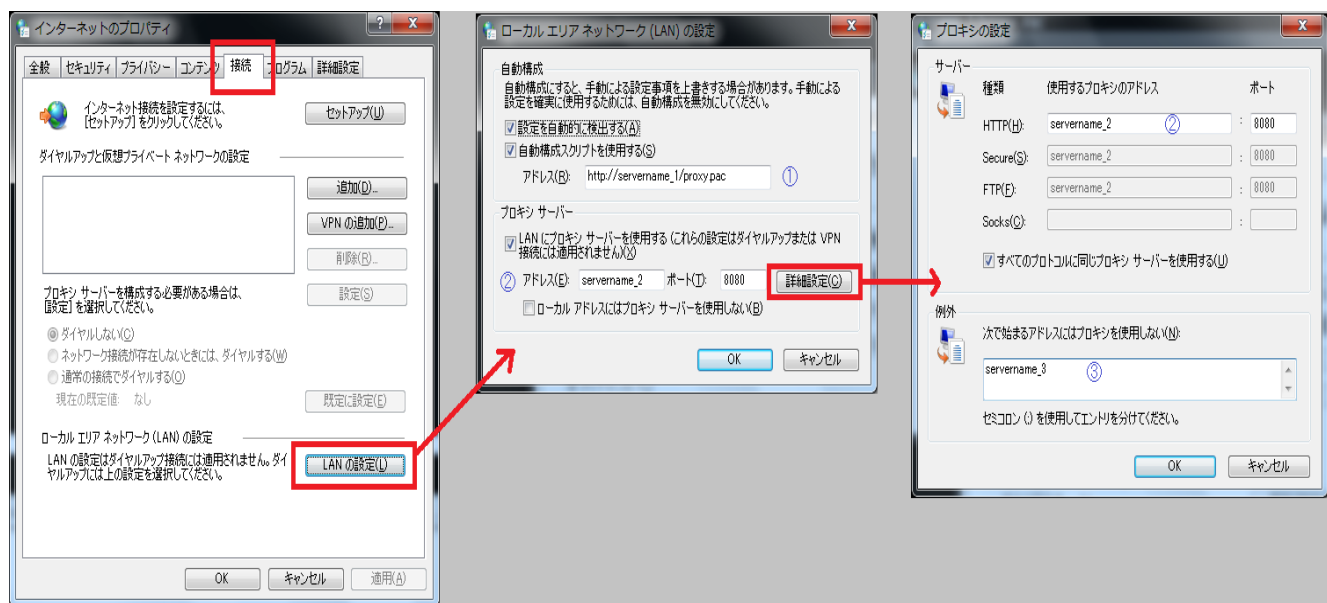
画面に表示されている証書シリアル番号がライセンス証書の記載通りか、有効期間が当日を含んでいるか、ライセンスされた CPU コア総数の上限がライセンス証書の記載通りかといった点をご確認の上、「閉じる」ボタンを押して下さい。

「閉じる」ボタンが押されたタイミングでライセンスファイルが下記の場所に作成されます。

```
%ALLUSERSPROFILE%¥PLUMSIX¥mtu_4.2¥license.dat
```

コンソール上へ次のように表示され、有効期間や CPU コア数の情報を知ることができます。

```
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
Windows 7 Professional Service Pack 1 (build 7601), 64-bit native
Number of NUMA nodes: 1
Number of physical processor packages: 1
Number of processor cores: 2
Number of logical processors: 2
Number of processor L1/L2/L3 caches: 4/1/0
Information - 有償ライセンス認証を完了しました。
有効期間は 2012-05-03 から 9999-12-31 まで
現在の CPU コア数=2, 許諾されている CPU コア数=4
```

9.1.9. プロキシー情報の構成

プロキシー情報の構成では Windows の「コントロールパネル」にある「インターネットオプション」の有効な設定があれば引用されます。

たとえば上記のように構成されている場合、「申請データの送信（自動）」画面が開くと上記設定を引用して初期表示が右図のようになります。丸数字の1から3までがそれぞれの画面で対応する個所を示しています。

インターネットオプション構成内容を引用せず、ライセンス認証ウイザード内の一時的な設定を行うこともできます。その場合は「このセッションでのみ有効なプロキシ設定を一時的に使用す8る」をチェックしてその内容を、書き込み可能になった下のエディットボックスへ記述することができます。右図を参照してください。

9.2. 登録抹消

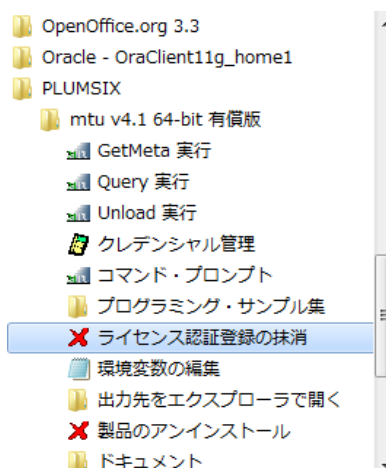
認証登録（52 頁）が完了すると、登録に使用したライセンス証書は、認証に使われたコンピュータと1対1に関連付けされます。関連付けデータは、弊社のライセンス認証サーバに保存されます。

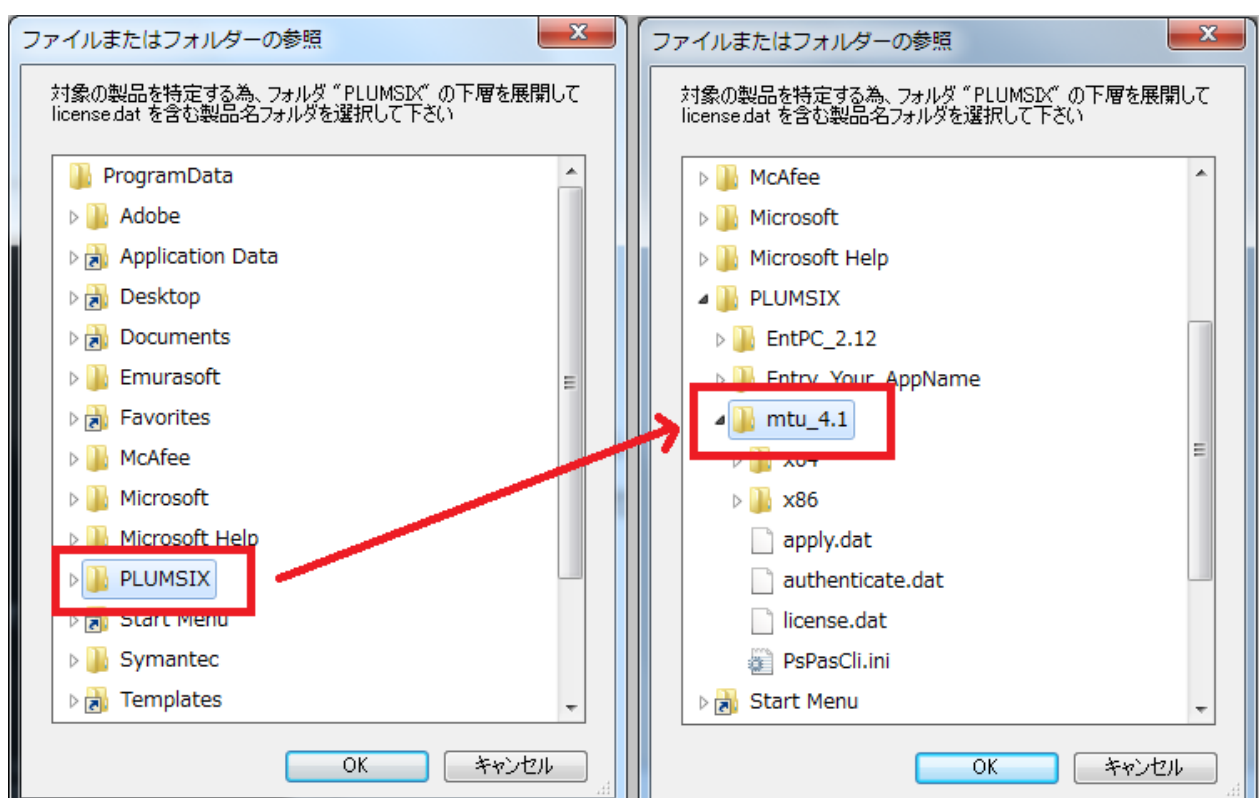
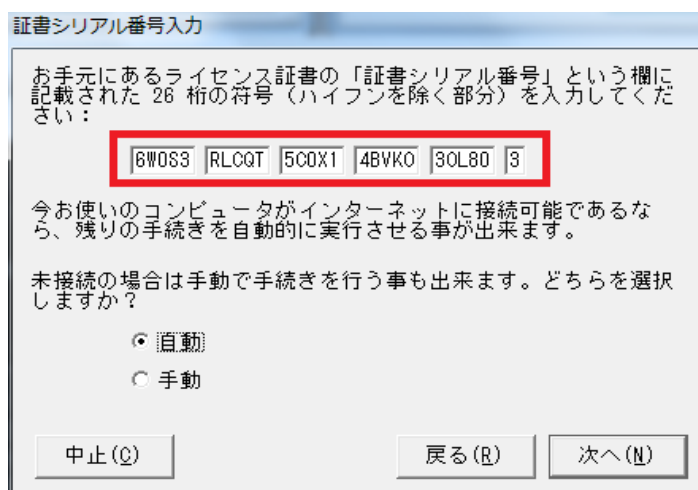
同じライセンスを別なコンピュータへ移動して使用したい場合はこの節に書かれている手順に沿って操作し、関連付けの登録を抹消してください。

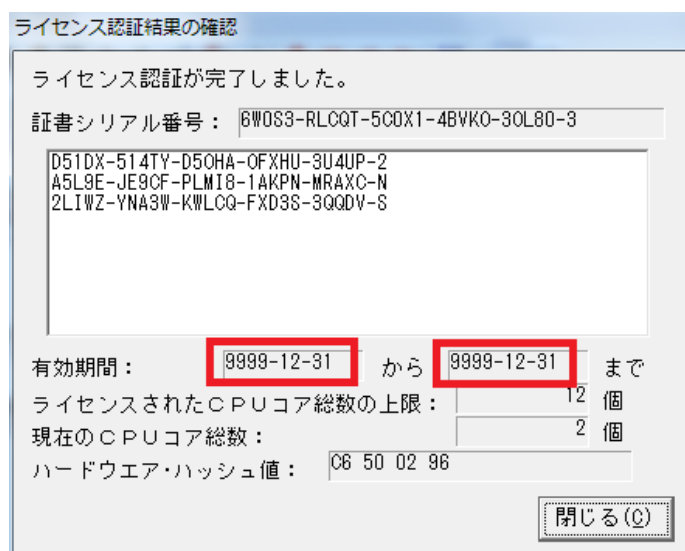
認証登録場合と同じ様に、コンピュータがインターネットへ接続可能な場合は、自動認証登録抹消が可能です。インターネットへ接続出来ない場合は、手動認証登録抹消を選択して下さい。

登録抹消が完了すると、再び別なコンピュータ上で「9.1.認証登録（52 頁）」に書かれている手順に従って再びライセンス認証ができるようになります。

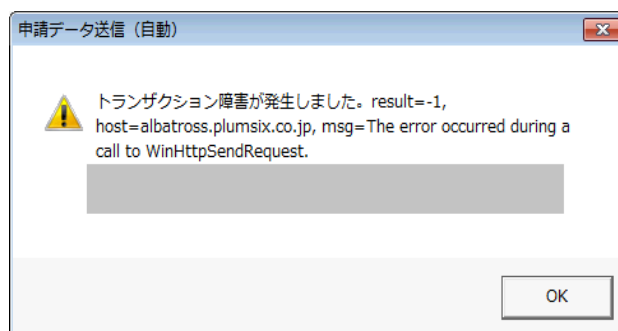
9.2.1. 「ライセンス認証登録の抹消」ツールの起動

最初に製品のメニューから「ライセンス認証登録の抹消」を選びます。

次に、「ファイルまたはフォルダの参照」というダイアログボックスが開きますので、初期選択の「PLUMSIX」というフォルダの1つ下層にある、ライセンス登録抹消対象の製品名のフォルダ((mtu_4.2)を選択ください。

「mtu_4.2」を選択した後、「OK」ボタンを押します。

「証書シリアル番号入力」画面が表示されます。使用している証書シリアル番号が初期表示されます。この点を除いて

「9.1.2.証書シリアル番号入力（53 頁）」から
「9.1.8.ライセンス認証結果の確認（60 頁）」まで

の間に書かれている手順と同様の操作を行ってください。

ライセンス認証登録の抹消が完了するとウイザードの最終画面で次のように表示されます。

「有効期間」の開始年月日が終了年月日と同じ値になります。これで登録抹消が完了したことの識別になります。

同時に弊社のライセンス認証サーバから、ライセンス証書と認証に使われたコンピュータの関連付けデータが取り除かれます。

9.3. ライセンス認証異常終了時の対応

9.3.1. トランザクション障害が発生しました。

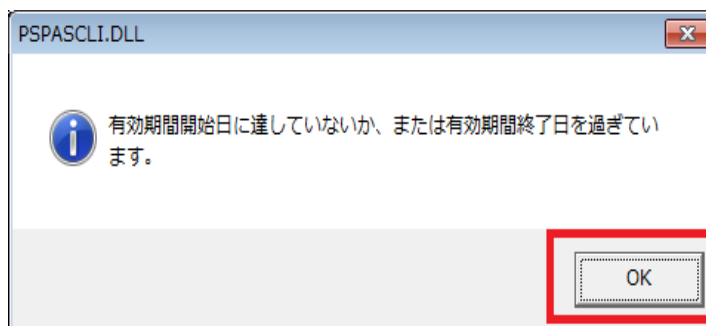
ライセンス認証サーバー障害が発生しています。「33.製品に関する問合せ先（194 頁）」にご案内しております連絡先までご連絡下さい。	

9.4. 評価ライセンスから有償ライセンスへの移行

評価ライセンスを有償ライセンスへ切替えるのはいつでも可能です。また、製品パッケージを再インストールすることなく移行作業を行うことができるので、評価期間中に残した作業の成果をそのまま引き継ぐことが可能です。

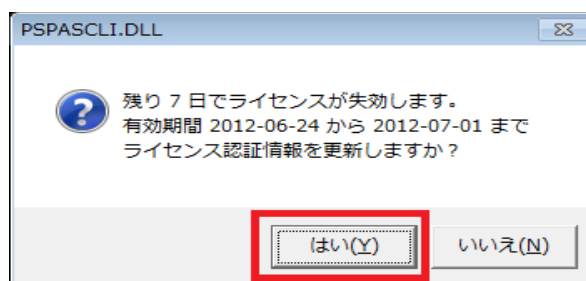
移行時期は大きく3つに分かれます。それぞれの時期で移行作業を開始する操作が異なりますので、分けて解説します。

9.4.1. 評価期限切れ

弊社から提供された評価ライセンスには予め利用可能な期間が設定されている為、製品が起動された日時がその範囲を超えていると、評価期間外にあると判定され右図にあるようなダイアログボックスが表示されます。

ここで「OK」を押すと、ライセンス認証ウイザードが表示されライセンス認証を行うことができるようになります。ウイザードの操作方法については「9.1.認証登録（52 頁）」にある記事を参照してください。

9.4.2. 期限切れ警告期間中

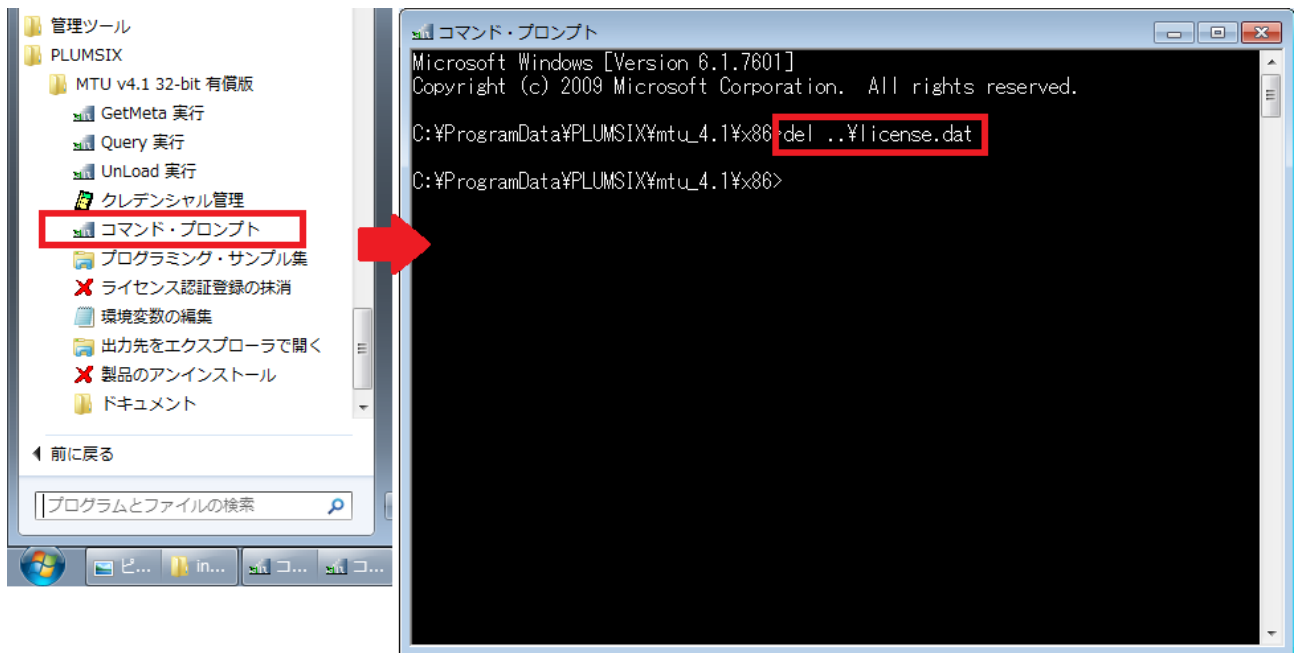
評価期間の終了日の 7 日前を過ぎると、評価期間がもうじき終わることを知らせる右図に示したダイアログボックスが表示されます。

ここで「はい」を選択するとライセンス認証ウイザードが表示されライセンス認証を行うことができるようになります。ウイザードの操作方法については「9.1.認証登録（52 頁）」にある記事を参照してください。

「いいえ」を選択した場合は、そのまま処理を続行することができます。日付が替わるまでダイアログボックスは表示されません。

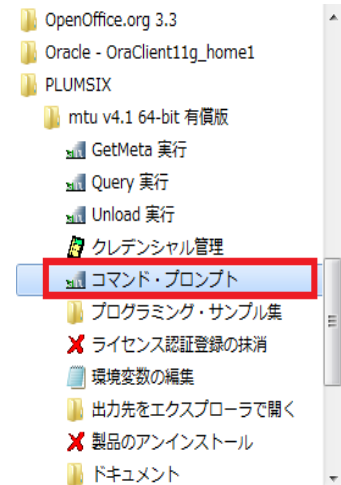
9.4.3. 期限切れ警告期間前

期限切れの警告を受取る前にライセンスを移行することも可能です。製品のメニューから「コマンドプロンプト」を選択してください。下図は「コマンド・プロンプト」のある場所を示しています。コマンド・プロンプトを開いたら、プロンプトへ

```
del ..¥license.dat
```

と入力します。この操作で評価ライセンス用のライセンスファイルが削除される為、次回起動時からは改めてライセンスファイルを作成する為、自動的にライセンス認証ウイザードが起動します。ウイザードの操作方法については「9.1. 認証登録(52 頁)」にある記事を参照してください。

10. アンインストール

製品のアンインストールを開始する前に Administrators グループに所属するユーザでログインして下さい。

アンインストールに当たっては「第1の方法」か「第2の方法」のどちらかを選び実施した後、「残りの手順」を実施してください。どちらの手順もコマンドプロンプトを使用しますので製品メニューに含まれる「コマンド・プロンプト」を選択してください。

10.1. 第1の方法（選択）

インストール時に自動的に作成された _connect.sql に含まれている CONNECT 文の接続先スキーマがアンインストール対象のスキーマ（所有ユーザ）であることを確認します。異なっている場合は編集し保存して下さい。その後、製品フォルダにある UnInstall.bat を実行すれば製品コンポーネントを構成するデータベース・オブジェクトが削除されます。以下に UnInstall.bat を実行した例を示します。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64>uninstall

SQL*Plus: Release 11.2.0.1.0 Production on 金 3月 9 16:41:03 2012

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

rem 以下のステートメントは INSTALL 時に指定されていた接続先です。
connect SYSTEM/xxxxxxx@xxxxxxxxxxxxxx;
接続されました。
drop PACKAGE APLDDP01;

パッケージが削除されました。

drop TABLE APLDDT01;

表が削除されました。

drop TABLE APLDDT02;

表が削除されました。

drop TABLE APLDDT03;
```

```
表が削除されました。

drop TABLE APLDDT04;

表が削除されました。

exit;
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options との接続が切断されました。
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64>
```

10.2. 第2の方法（選択）

UnInstall.bat の実行に替わる別な方法として強制的にデータベース上の全ての製品コンポーヘントを一括削除することも出来ます。製品フォルダ¥input¥drop_all_apldd.sql を DBA ロールを付与されたユーザで実行してください。以下に drop_all_apldd.sql を実行した例を示します。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64>sqlplus SYSTEM/xxxxxxx@xxxxxxxxxxxxxx

SQL*Plus: Release 11.2.0.1.0 Production on 金 3月 9 16:43:11 2012

Copyright (c) 1982, 2010, Oracle.  All rights reserved.



Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options
に接続されました。
SQL> @input¥drop_all_apldd.sql
PACKAGE "SYSTEM"."APLDDP01" was dropped.
TABLE "SYSTEM"."APLDDT01" was dropped.
TABLE "SYSTEM"."APLDDT02" was dropped.
TABLE "SYSTEM"."APLDDT03" was dropped.
TABLE "SYSTEM"."APLDDT04" was dropped.

PL/SQL プロシージャが正常に完了しました。

製品コンポーネントの削除が終わりました。SQL*Plus を終了するには exit を入力してください。
SQL> exit
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options との接続が切断されました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64>
```

10.3. 残りの手順（必須）

1. 出力先フォルダ内のファイルと出力先フォルダは必要に応じて手動で削除して下さい。

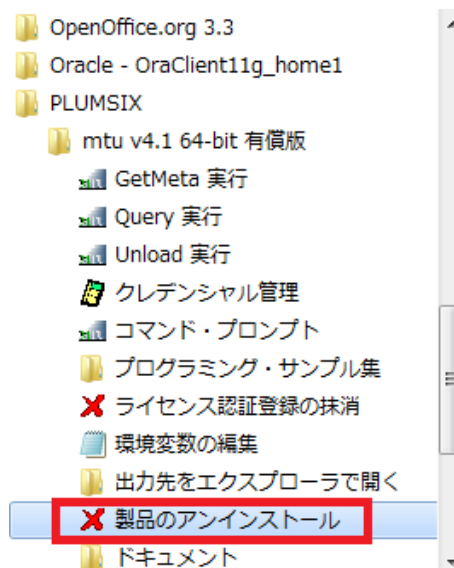
2. 製品フォルダ内(頁)のファイル、サブ・フォルダ、及び製品フォルダは必要に応じてバックアップし、不要であれば手動で削除して下さい。

   製品フォルダ内(43 頁)は、インストール後初めて製品を起動すると作成されます。このデータは製品のアンインストールでは削除されません。不要であれば手作業で削除してください。

3. インストール作業で付与した所有ユーザの Oracle システム権限を取り消します。

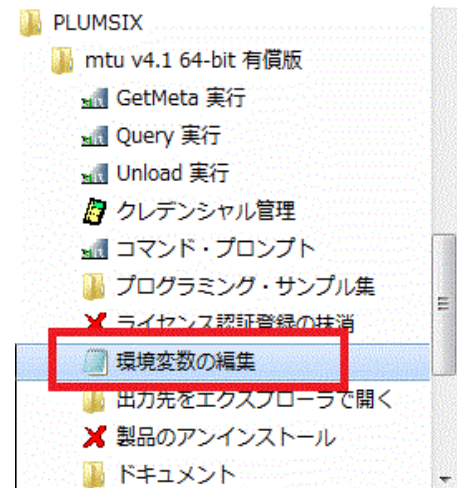
4. 所有者ユーザ用として使用したスキーマが不要であれば削除して下さい。

5. 別なコンピュータへライセンスを移動して引き続きご利用の場合は「9.2.登録抹消(62 頁)」にある手順に沿ってライセンス認証登録を抹消してください。

6. 製品メニューに含まれている「製品のアンインストール」を選択して製品をアンインストールしてください。

7. 以上で終了です。

11. 環境変数について

11.1. 環境変数の編集方法

環境変数は製品フォルダ[15]にあるEnv.batスクリプトを介して設定します。直接メモ帳などのテキストエディタを使って開くか、右図のように製品メニューに用意されている「環境変数の編集」というショートカットを利用して開くこともできます。

11.2. 環境変数を使うときの留意点

11.2.1. 「任意文字列」という値について

選択肢の中で「任意文字列」と書いてある場合は、環境変数へ文字列が設定されているか、そうでないかがテストされます。仮に下記のような設定をした場合は同じ意味として扱われることにご注意ください。true と false という単語の指す意味は動作に関係しません。

```
set <環境変数名>=true
set <環境変数名>=false
```

上記とは反対の意味を指定したい場合には下記のようにしてください。

```
set <環境変数名>=
```

11.2.2. 環境変数の値として使用できない文字について

一般的な注意点として、コマンド・インタプリタがパイプ処理やリダイレクト処理、あるいはバックグラウンドプロセスとしてコマンドを実行する指定をする為の文字は環境変数の値として指定できません。例)"|"、">"、"<"、"&&"

また環境変数の値で置換することを示す文字を指定すると誤動作の原因となります。例)Windows では "%"、C Shell, Boune Shell では "$"

[15] %ALLUSERSPROFILE%¥PLUMSIX¥mtu_4.2¥x86、または%ALLUSERSPROFILE%¥PLUMSIX¥mtu_4.2¥x64

11.2.3. 文字種、文字数の制限について

製品が使用する環境変数には、ファイル名、ディレクトリ名、あるいは Oracle データベース上のオブジェクト名を指定するものがあります。それらへ、OS や Oracle の制限によって指定することが出来ない文字や最大文字数の制限を超えて指定すると、誤動作の原因となりますのでご注意ください。OS や Oracle の制限についてはそれぞれの製品のマニュアルやサポート窓口にてご確認ください。

11.3. エスケープ文字

Windows プラットホームでは環境変数 LISTTABLE（84 頁）, LISTFIXED（84 頁）, 及び LISTEXCPT（84 頁） にワイルドカード検索文字「%」を指定する場合は前方にもう一つ「%」を付加してエスケープして下さい。

- （例）TBL01 , TBL02, TBL03 の3表があってその全てを指定する場合。

```
LISTEXCPT=TBL%%
```

LISTTABLE, LISTFIXED, 及び LISTEXCPT に名前に「_」を含む表を指定したい場合、「¥」を付加してエスケープしてください。

- （例）URIAGE_200110 を指定する場合。

```
LISTEXCPT=URIAGE¥_200110
```

11.4. 各機能に対する影響

「●」印は当該環境変数が機能で使用されていることを意味します。

表 14 環境変数と各機能の関係(ABC 順)

環境変数名	UnLoad	Query	GetMeta	ConsDiSable	ConsEnable	GetFkrb	GetTblP	GetTblN	GetPlsP	GetPlsN	InsTall	出荷時設定
ALTER_DEST	●	●										≪無≫
AUXILIARY_DEST	●	●										≪無≫
BIND_SIZE	●	●										20M
BULK_SIZE	●	●	●			●	●	●	●	●		1000
CHARSETFORM	●	●										≪無≫
CHARSETID	●	●										0
CLIENT_ID	●	●	●	●	●	●	●	●	●	●	●	MTU_CLIENT

環境変数名	UnLoad	Query	GetMeta	ConsDiSable	ConsEnable	GetFkrb	GetTblP	GetTblN	GetPlsP	GetPlsN	InsTall	出荷時設定
CMNTLVL			●									1
COMPRESS			●									True
CONSISTENT	●	●										True
DATAEXT	●	●										≪無≫
DATE_MASK	●	●	●	●	●	●	●	●	●	●	●	YYYYMMDDHH24MISS
DB_FILE_MULTIBLOCK_READ_COUNT	●	●										128
DEPENDENT			●									true
DFILE_ALT_DIRS	●	●										≪無≫
DIRALIAS	●	●										ALIAS1
DISCRETE	●	●										true
DST_INFO	●	●										scott/tiger
EMBED_COLUMN_NAME	●	●										≪無≫
ENCLOSER	●	●										~~
ENCLOSER2	●	●										≪無≫
ENDTERM	●	●										true
EVENTS	●	●	●	●	●	●	●	●	●	●	●	≪無≫
EXEC_LOAD	●	●										sqlldr
EXEC_PLUS	●											sqlplus
EXTNAMEBLOB	●	●										blo
EXTNAMECLOB	●	●										clo
EXTNAMESQL		●										sql
FBQSCN	●	●										≪無≫
FBQTIME	●	●										≪無≫
FEEDBACK	●	●	●			●	●	●	●	●		10
FILEBIND		●										bind.dat
FILEEXCPT	●					●	●	●				excpt.dat
FILEFIXED	●					●	●	●				fixed.dat
FILEFKRB						●						fkrb.sql
FILELOG	●	●	●	●	●	●	●	●	●	●	●	≪無≫
FILEMETA			●									meta.sql
FILEPLSN										●		plsn.txt
FILEPLSP									●			plsp.txt
FILESIZE	●	●										2000M
FILETABLE	●					●	●	●				table.dat
FILETBLN								●				tbln.txt
FILETBLP							●					tblp.txt
FILETRACE	●	●	●	●	●	●	●	●	●	●	●	_trace.log
FINDSTRCMD	●	●										≪無≫
FPNUMFMT	●	●										≪無≫
GETEXTTBL	●	●										≪無≫
INPUT	●	●				●					●	%MTU_USER_DIR%input
IO_OVERLAP_SCALE	●	●										1
LISTEXCPT	●					●	●	●				≪無≫
LISTFIXED	●					●	●	●				≪無≫

環境変数名	UnLoad	Query	GetMeta	ConsDiSable	ConsEnable	GetFkrb	GetTblP	GetTblN	GetPlsP	GetPlsN	InsTall	出荷時設定
LISTTABLE	●					●	●	●				≪無≫
LOADMETHOD	●	●										TRUNCATE
LONGTRANSIT	●	●										0
MAXLONGSIZE	●	●										1000
MERGE_LOBS_INTO_SDF	●	●										≪無≫
METADIV			●									≪無≫
MIN_BULK_SIZE	●	●	●			●	●	●	●	●		100
MINIDUMP_TYPE	●	●	●	●	●	●	●	●	●	●	●	0x0
MINIMUM_FREE_SIZE	●	●										1024M
MNGSCROUT	●											0xec00
NMENCLS	●	●	●			●						true
NOBYTEORDERMARK	●	●										≪無≫
NUM_ASYNCLOBS	●	●										%PARALLELISM%
NUM_ROWS	●											1
NUMBER2BCD	●	●										≪無≫
NUMDAYS_AUDIT			●									≪無≫
OPTIMIZER_HINTS	●											≪無≫
OPTIMIZER_MODE	●	●	●	●	●	●	●	●	●	●	●	≪無≫
OUTPUT	●	●	●			●	●	●	●	●		%MTU_USER_DIR%output
OVERLAP_BUFFER_LENGTH	●	●										1M
PARALLELISM	●	●										4
PARTITIONING	●	●										true
PRINTCOLID	●		●			●						true
PRIORITY_CLASS	●	●	●	●	●	●	●	●	●	●	●	BELOW_NORMAL
PRE_REP_EXEC_PLS	●	●										≪無≫
POST_REP_EXEC_PLS	●	●										≪無≫
QUERYFIX		●										qryfix
QUERYVAR		●										qryvar
QUERYFILENAME		●										*
RECLENGTH	●	●										true
RECOVERABLE	●	●										true
RERUNPOINT	●	●										≪無≫
REVERSEBYTEORDER	●	●										≪無≫
ROWID_SPLIT_MIN_SIZE	●	●										0M
ROWID_SPLIT_NUM_PARTS	●	●										4
ROWS	●											true
ROWSEPARATOR	●	●										≪無≫
SERVEROUTPUT	●	●	●	●	●	●	●	●	●	●	●	≪無≫
SHOWCLTDTM	●	●	●	●	●	●	●	●	●	●	●	
SHOWSTAT	●	●	●	●	●	●	●	●	●	●	●	≪無≫
SHOWSTMT	●	●										true
SHOWVERS	●	●	●	●	●	●	●	●	●	●	●	≪無≫
SKIPSCR	●	●										≪無≫
SORTING	●											≪無≫
SORT_AREA_SIZE	●	●	●	●	●	●	●	●	●	●	●	≪無≫

環境変数名	UnLoad	Query	GetMeta	ConsDiSable	ConsEnable	GetFkrb	GetTblP	GetTblN	GetPlsP	GetPlsN	InsTall	出荷時設定
SQLFMT			●									0x8000
SRC_USER	●	●	●	●	●	●	●	●	●	●	●	scott
STDOUT	●	●										0x0000
STREAM_LOCATOR	●	●										≪無≫
TAF_DELAY_TIME	●	●										60
TERMINATOR	●	●										,
TIMESTAMP_MASK	●	●	●	●	●	●	●	●	●	●	●	YYYYMMDDHH24MISSXFF
TIMESTAMP_TZ_MASK	●	●	●	●	●	●	●	●	●	●	●	YYYYMMDDHH24MISSXFFTZHTZM
THREAD_PRIORITY	●	●	●	●	●	●	●	●	●	●	●	BELOW_NORMAL
USERID	●	●	●	●	●	●	●	●	●	●	●	SCOTT/TIGER
USUALPATH	●	●										true
WINTARGET	●	●										true

11.5. 環境変数と値の意味

表 15 環境変数と値の意味(ABC 順)

環境変数名／設定可能な値の範囲と意味	省略時初期値
ALTER_DEST	≪無≫
DST_INFO で指定した接続先と異なるスキーマへデータをリロードする際にその名前を指定します。セキュリティー上の理由等でユーザのログインを認めていないスキーマへのリロードを DBA 等の特権者が代行する時に使用します。 ここに指定した名前は、制御ファイルのテーブル名を修飾し、_conn.sql に追加される alter session set current_schema の 引数として使用されます。 ALTER_DEST の指し示す値が実際に評価されるのは、アンロードしたデータを再ロードするときです。MTU は接続先 DB にここで示したスキーマがあるかはチェックしません。再ロード先データベースがアンロード対象データベースであるとは限らないからです。	
AUXILIARY_DEST	≪無≫
≪無≫: プライマリ・データ・ファイルの出力先を分散しません。 有効なファイルシステム上のディレクトリ名を1つ指定するか、2つ以上のディレクトリ名をカンマ区切りリストで指定：アンロード時、または再ロード時に I/O 競合を抑え所要時間を短縮したい場合に指定します。指定されたディレクトリが独立して動作するデバイス上に設けられたファイルシステムである時、I/O が分散されパフォーマンスの向上効果があります。 OUTPUT と同様に C:¥Documents and Settings¥... といった途中に空白が含まれるディレクトリ名も指定することが出来ます。この時、二重引用符でディレクトリ名を囲む必要はありません。UNC 表記法に基づいたネットワー	

環境変数名／設定可能な値の範囲と意味	省略時初期値
ク上の共有ディレクトリ名を指定することも出来ます。 実在しないディレクトリ名が１つ以上指定されたときは、エラーメッセージを伴って異常終了します。	
BIND_SIZE	655360
16384 ≦ BIND_SIZE ≦ 符号なし32ビット整数の最大値 ホスト配列変数と出力バッファへ割り当てるメモリの最大サイズを指定します。この値をホスト配列変数の出力バッファ1行当りに必要なバイト数と環境変数 PARALLELISM（90 頁）の積で割った数を求め、環境変数 BULK_SIZE（75 頁）へ指定した値と比較します。BULK_SIZE の方が大きければ割った値を要素数とします。 反対に BULK_SIZE の方が小さければ設定値を要素数とします。16kB 未満の値を指定するとエラーになります。補助単位として k,M,G,T,P を数値の後に追加して使うこともできます。	
BULK_SIZE	100
1 ≦ BULK_SIZE ≦ 符号なし32ビット整数の最大値 ホスト配列変数の最大の要素数を指定します。実際の要素数はこの値に指定した数を上限に環境変数 BIND_SIZE（75 頁）と環境変数 PARALLELISM（90 頁）から割当可能な数を求めて決められます。 300 から 2000 までの範囲で概ね良好な性能が得られます。システムで使用可能な空きメモリのサイズによって指定可能な上限が異なります。要求された要素数に必要なメモリを確保できなかった場合はエラーになります。 1 未満の値を指定するとエラーになります。	
CHARSETFORM	1
1: クライアントキャラクタセットへ変換 2: 各国語キャラクタセット（AL16UTF16）のまま無変換 NCHAR 型/ NVARCHAR2 型で各様されたデータの変換方法を切り替えることが出来ます。 何も値が指定されなければ、1 が指定されたものとみなします。 クライアントキャラクタセットの場合、環境変数 CHARSETID や NLS_LANG で上書きしなければ日本語 Windows 用 Oracle Database Client のデフォルトである JA16SJISTILDE が選択されます。 2 を選択した場合、データファイル以外のファイルは出力されなくなる、という制限があります。。これは NCHAR/NVARCHAR2 以外のデータ型によって出力された ASCII 文字を含んでいる、複数の文字エンコーディングが混在したデータファイルを再ロードするのに適した SQL*Loader 制御ファイルのグラマーが存在しない為です。 2 を選択すると上記制限の警告を行うため画面へ下記のように表示されます。 `Warning - 環境変数 CHARSETFORM へ値 2 が指定されました。移植不可能なデータ形式が選択された為、データ・ファイル以外は作成されません。`	
CHARSETID	0
0: データ・ファイル、CLOB 型の LOB ファイルのエンコーディングに環境変数 NLS_LANG に指定されたキャラクタ・セット（第３セグメント部分）を使用します。日本語 Windows を使用している場合には 832 を指定したのと同じ意味になります。 873 または 871: データ・ファイル、CLOB 型の LOB ファイルのエンコーディングに UTF-8 を使用します。この設定内容の影響を受ける Oracle データ型については「表 2 UnLoad 可能な Oracle データ型（15 頁）」、「13.Unicode 対応につ	

環境変数名／設定可能な値の範囲と意味	省略時初期値

いて (125頁)」にある記事を参照して下さい。

プラットフォームがサポートしている CHARSETID については、下記の問い合わせを行うことで調べることができます。

```
SELECT NLS_CHARSET_ID(VALUE) AS CHARSETID, SUBSTRB(VALUE,1,10) AS CHARSET_NAME
FROM V$NLS_VALID_VALUES WHERE PARAMETER = 'CHARACTERSET';
```

CLIENT_ID

省略時初期値: PS_OCI_CLIENT

≪無≫:デフォルトのクライアント識別子を使用します。

任意文字列:64 バイト以下の文字列を設定すると v$active_session_history 表 の CLIENT_ID 列、v$session 表の CLIENT_IDENTIFIER 列、SQL trace ファイル中の CLIENT ID タグ等の各種診断出力へ、ラベル付けする事が出来るようになります。

Oracle Database 製品マニュアル「パフォーマンス・チューニング・ガイド」「21.1 End to End Application Tracing」にて紹介されているツールを使用した診断出力の分析役立てる事ができます。CLIENT ID タグを利用すると、製品に限定してトレースを収集できるようになるので、診断の手間が省かれます。

CMNTLVL

省略時初期値: 0

0: 表／列コメント定義 SQL 文を出力しません。

1: 登録済の表／列コメントについてだけ定義 SQL 文を出力します。

2: 表／列コメントが未登録なら空の定義 SQL 文を出力します。

COMPRESS

省略時初期値: ≪無≫

≪無≫: 追加エクステントを初期エクステントにまとめません。COMPLESS オプション付きで export ユーティリティーを使用した結果、初期エクステントサイズが必要以上に大きくなってしまった場合などに使用してください。

任意文字列: 追加エクステントを初期エクステントにまとめます。

※DBA_SEGMENTS/USER_SEGMENTS の下記の列の値が使用されます。

パラメータ	≪無≫	任意文字列
initial	initial_extent	bytes
next	initial_extent	next_extent
minextents	min_extents	1

CONSISTENT

省略時初期値: ≪無≫

≪無≫: 文レベルの読取一貫性を保証します。

任意文字列: 下記の画面表示が出力された後から全てのデータ読取が完了するまでの間、トランザクションレベルの読取一貫性を保証します。

```
データ・ファイルと制御ファイルを出力中です。output
```

※分離レベルを Repeatable read にしたい場合には任意文字列を指定して下さい。但し副作用として「ORA-01555: ス

<table>
<tr><th>環境変数名／設定可能な値の範囲と意味</th><th>省略時初期値</th></tr>
<tr><td colspan="2">ナップショットが古すぎます」が発生し易くなります。</td></tr>
<tr><td>DATAEXT</td><td>dat</td></tr>
<tr><td colspan="2">任意文字列: CSV/固定長データ・ファイルの拡張子を指定します。</td></tr>
<tr><td>DATE_MASK</td><td>YYYYMMDDHH24MISS</td></tr>
<tr><td colspan="2">日付データ型の書式マスク文字列を指定します。</td></tr>
<tr><td>DEPENDENT</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫: 表の定義順序を表名の昇順とします。参照整合性制約によりキーのデータ型の記述が省略可能であってもデータ型が常に記述されます。
参照整合性制約の定義は全ての表定義の後方にある″表名_FK″スクリプト内で ALTER TABLE ADD CONSTRAINT 制約名 FOREIGN KEY 列リスト REFERENCES 親表名 コマンドで定義されます。
任意文字列: 子表の参照整合性制約によって参照される親表が子表よりも前方に定義されます。外部キーについてはデータ型の記述が省略されます。参照整合性制約は表定義の中に定義されます。</td></tr>
<tr><td>DFILE_ALT_DIRS</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫: 空き領域不足が発生してもフェイルオーバーしません。
カンマ区切りリスト: 記述されている要素順に代替出力先として空き領域が十分あるかどうかが試され、十分であればフェイルオーバーします。空き領域サイズは環境変数 MINIMUM_FREE_SIZE(86 頁)で指定します。
EMBED_COLUMN_NAME(78 頁)が設定されている場合、設定内容は無視されます。領域不足を検出しても代替先へのフェイル・オーバーはしません。そのまま、自発的に異常終了します。
詳しい内容については「18.領域あふれ時のフェイルオーバー(139 頁)」の記事を参照してください。</td></tr>
<tr><td>DIRALIAS</td><td>≪省略不可≫</td></tr>
<tr><td colspan="2">任意文字列: ロード予定先DBに実在するディレクトリ・オブジェクトを指定します。ディレクトリ・オブジェクトが無いときは create or replace directory ディレクトリ・オブジェクト名 as 'ディレクトリのパス' を実行して作成し、grant read, write on directory ディレクトリ・オブジェクト名 to ロールまたはユーザ名 を実行してオブジェクト権限を付与してください。
パスにアルファベットが含まれる場合、大文字と小文字が区別されるので注意して下さい。LONGTRANSIT=2 の時に必須です。
アンロードするDBにこのパラメータで指定したディレクトリ・オブジェクトが無くてもエラーにはなりません。</td></tr>
<tr><td>DISCRETE</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫: 制御ファイルへ CSV/固定長データを含めます。制御ファイルに BEGINDATA 句と CSV/固定長データが追加されます。*.dat ファイルは出力されません。
任意文字列: 制御ファイルと CSV/固定長データ・ファイルを分離します。SQL*Loader の制約上、この変数に文字列を指定しない場合は環境変数 RECLENGTH(93 頁)にも文字列を指定しないで下さい。</td></tr>
</table>

環境変数名／設定可能な値の範囲と意味	省略時初期値
DST_INFO	≪省略不可≫

ロード予定先スキーマへの接続文字列を指定します。

書式は

<ユーザ名>/<パスワード>@<接続識別子>

UnLoad/Query 機能によって出力された、_conn.sql ファイルの CONNECT コマンドのパラメータ、_par0.txt/_par1.txt ファイルの USERID パラメータへ設定されます。

上記の書式以外にも先頭1文字目に半角アットマーク' @'で始まるファイル名を設定すると、資格証明情報ファイルから認証情報を入力する事もできます。詳しくは「15.パスワードの秘匿化について(128 頁)」に書かれた記事を参照してください。

192 バイト以下の文字列を設定する事ができます。

EMBED_COLUMN_NAME	

≪無≫：可変長のデータ・ファイルを出力する際にカラム名を表示するデータをデータ・ファイルの1行目に出力しません。

任意文字列：カラム名データを出力します。それに加えて行データ内の列値区切を指定することが出来ます。区切は半角コロン文字で区切られた3つの文字列セグメントで表現されます。データ型の表現に使用できる区切り文字は Windows のコマンド・インタプリタの正常な動作に影響を与えるもの('>','<','|',';','%' など)を除いて自由に選ぶことが出来ます。半角コロン文字で区切られた各セグメントは下記の意味づけと使用可能な文字列があります。

セグメント	意味	使用可能な文字列の例
1 (必須)	列区切り	0x09 (TAB), 0x2c (','), 0x7c ('|')
2 (省略可)	文字列型の区切り開始	0x22 (2重引用符), 0x27 (単一引用符),0x28 ('('), 0x29 (')') など。
3 (省略可)	文字列型の区切り終了	

第2セグメントに区切り文字が設定されている場合、第3セグメントを省略する事が出来ます。この場合、第 3 セグメントには第 2 セグメントに指定されたものと同一の文字列が使用されます。半角コロンは省略しないで下さい。ここで指定した設定内容は環境変数 TERMINATOR(98 頁)、ENCLOSER, ENCLOSER2(78 頁)の設定内容よりも優先されます。

例えば Microsoft Excel で加工無しでそのまま copy & paste 可能なデータ表現は下記のような設定で出力することが出来ます。

set EMBED_COLUMN_NAME=0x09:0x22:

注意：EMBED_COLUMN_NAME が指定されている場合、AUXILIARY_DEST(74 頁)、DFILE_ALT_DIRS(77 頁)、RECLENGTH(93 頁)、ENDTERM(79 頁)の値は≪無≫とみなされます。反対に DISCRETE(77 頁)は任意文字列が指定されたものとみなされます。

カラム名にマルチバイト文字が含まれている場合、出力される文字エンコーディングが CHARSETID(75 頁)の設定値に適したものが自動的に選択されます。つまり、CHARSETID が 873 や 871 などの utf-8 系であれば、カラム名のマルチバイト文字も utf-8 で出力されます。

ENCLOSER, ENCLOSER2	≪無≫

環境変数名／設定可能な値の範囲と意味	省略時初期値
文字列型と日付型データを囲む為の文字、または文字列を指定します。 列値に STREAM_LOCATOR（98 頁）で指定した文字や文字列が含まれる場合は必ず指定して下さい。 通常は「”」（半角二重引用符文字）を指定しますが、列値に「”」自身が含まれている場合には別な文字、あるいは文字列を指定してください。例えば「~~」（チルダ記号を二つ）という文字列を指定するとほとんどの場合に対応できます。ただし文字列を指定するためには Oracle8i 以上のの SQL*Loader が必要です。0x を前置した 16 進数表現の文字、文字列を使用する事も出来ます。 ENCLOSER2 に文字や文字列を指定すると、文字列型データを囲む前方の文字／文字列とは異なる文字／文字列を後方の文字／文字列として使うことが出来ます。ENCLOSER2 を省略すると ENCLOSER へ指定したものと同じ文字／文字列を使うという意味になります。 例えば文字列型データを[mojimoji]という表現で出力したければ ENCLOSER へ[を ENCLOSER2 へ]を指定してください。	
ENDTERM	≪無≫
≪無≫: CSV または固定長データレコード末端に TERMINATOR へ指定された文字または文字列を出力しません。制御ファイルに TRAILING NULLCOLS オプションが追加されます。 任意文字列: 環境変数 STREAM_LOCATOR（98 頁）へ指定された文字または文字列を出力します。 SQL*Loader の制約上この変数に文字列を指定しない場合は環境変数 RECLENGTH（93 頁） にも文字列を指定しないで下さい。	
EVENTS	≪無≫
alter sesion set events 'STRING' 文の STRING に相当する文字列を設定するとイベントを設定できます。入力可能な文字数は 1 バイト以上 512 バイト以下です。イベントを設定すると MTU が使用しているシャドウ・プロセスの診断情報をトレースファイルへ出力することが出来ます。イベントには、正確な指示の下に設定しないとデータベースへ致命的はダメージを与える危険なものがあり、かつトレースファイルは診断出力情報を大量に含み Oracle サーバのディスクを消費することがあるので、特にサポートの指示が無い限りは値を指定しないでお使い下さい。	
EXEC_LOAD	≪省略不可≫
ロード実行時に使用する SQL*Loader の実行形式ファイル名を指定します。_make.bat に指定した文字列による呼出が出力されます。	
EXEC_PLUS	≪省略不可≫
管理作業用スクリプト適用時に使用する SQL*Plus の実行形式ファイル名を指定します。_make.bat に指定した文字列による呼出が出力されます。	
EXTNAMEBLOB	≪省略不可≫
BLOB 値をファイル出力するときの拡張子名を指定します。	
EXTNAMECLOB	≪省略不可≫
CLOB/NCLOB 値をファイル出力するときの拡張子名を指定します。	

<table>
<tr><th>環境変数名／設定可能な値の範囲と意味</th><th>省略時初期値</th></tr>
<tr><td>EXTNAMESQL</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫: 任意問合せファイル名の拡張子を限定しません。拡張子を問わず QUERYVAR(92 頁)、QUERYFIX(91 頁)に指定された場所にあるファイルをすべて問合せ対象の候補にします。
任意問い合わせが記述されたスクリプトファイルの拡張子名を限定します。QUERYVAR(92 頁)、QUERYFIX(91 頁)に指定された場所にあるファイルでもこの変数に指定した拡張子でないファイルは読み込まれません。</td></tr>
<tr><td>FBQSCN</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫: フラッシュバック問合せを無効にします。
内部的に DBMS_FLASHBACK.ENABLE_AT_SYSTEM_CHANGE_NUMBER を実行して SCN ベースのフラッシュバック目標をセットします。
0 ≦ FEEDBACK ≦ 符号付き64ビット整数の最大値: FBQTIME と共に設定された場合は FBQSCN(80 頁)に設定された値が優先されます。UNDO セグメント上から失われているような古すぎる変更前情報を得ようとすると異常終了します。</td></tr>
<tr><td>FBQTIME</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫: フラッシュバック問合せを無効にします。
書式モデル yyyymmddhh24miss で日付時刻を指定: 内部的に DBMS_FLASHBACK.ENABLE_AT_TIME を実行して日付時刻ベースのフラッシュバック目標をセットします。
UNDO セグメント上から失われているような古すぎる変更前情報を得ようとすると異常終了します。</td></tr>
<tr><td>FEEDBACK</td><td>10</td></tr>
<tr><td colspan="2">0 ≦ FEEDBACK ≦ 符号なし32ビット整数の最大値: 進捗表示の更新頻度を指定します。
0: 進捗表示をしない
1 以上: 進捗表示をする(数値が大きければ大きいほど進捗表示の更新頻度が少なくなります)
FILELOG(81 頁)を使って画面表示をファイルへリダイレクトしている場合は、この変数に指定する値を買えることによりログの量を調節することが出来ます。</td></tr>
<tr><td>FILEBIND</td><td>≪無≫</td></tr>
<tr><td colspan="2">バインド変数が定義されたファイル名か、直接バインド変数を定義した key=value 形式の表現をを指定します。
ファイルの位置は INPUT で指定します。このファイルにはバインド変数の定義情報の他、任意のコメント文字列を含めることが出来ます。コメント文字列は必ず「#」ではじめてください。コメント文字列は行頭から開始することも行の半ばから開始することも出来ます。
key=value 形式の場合、表現は半角カンマで区切って複数指定することが出来ます。下記にその事例を示します。
<code>set FILEBIND=:b_job="SALESMAN", :b_sal=1300, :b_hiredate="1980-01-01"</code>
例えば環境変数 QUERYFIX(91 頁)、QUERYVAR(92 頁)へ指定できる下記のような SELECT 文と組み合わせて使用することも可能です。</td></tr>
</table>

環境変数名／設定可能な値の範囲と意味	省略時初期値

`set FILEBIND=:b_job="MANAGER",:b_sal=2500`
`set QUERYVAR="select empno,ename,sal into LOW_SAL_MGR from emp where job = :b_job and sal < :b_sal;"`
 	
key=value 形式の文字列型 value を表現するための囲い文字として半角単一引用符の他に下記の組合せも使用可能です。
 	
`"", [], {}, (), <>`
 	
この環境変数を指定しない場合はバインド変数が無いものとみなします。	

FILEEXCPT	≪無≫
この変数に設定されたファイルに含まれるテーブル名を処理対象から除外します。ファイル内の各行はワイルドカード文字を使い複数のテーブルを指定することも可能です。実在しないテーブル名を記入しても無視されます。ファイルの位置は INPUT(83 頁)で指定します。このファイルにはテーブル名の他、任意のコメント文字列を含めることが出来ます。コメント文字列は必ず「#」ではじめてください。コメント文字列は行頭から開始することも行の半ばから開始することも出来ます。 実在しないファイル名を指定すると異常終了します。	
FILEFIXED	≪無≫
環境変数 LISTTABLE(84 頁)、FILETABLE(82 頁)に指定したテーブルのうち、固定長データ形式で出力させたいものを指定します。ワイルドカード文字を使い複数のテーブルを指定することも可能です。ファイルの位置は INPUT(83 頁)で指定します。このファイルにはテーブル名の他、任意のコメント文字列を含めることが出来ます。コメント文字列は必ず「#」ではじめてください。コメント文字列は行頭から開始することも行の半ばから開始することも出来ます。	
FILEFKRB	≪省略不可≫
GetFkrb 機能の出力ファイル名を設定します。	
FILELOG	≪無≫
≪無≫: CON, con: メッセージを画面へ表示します。 任意文字列: 画面に出力されるメッセージを格納するファイル名を指定します。ファイル名より前方のパスを省略した時の出力先ディレクトリは OUTPUT(89 頁)へ設定した先です。	
FILEMETA	≪省略不可≫
GetMeta 機能の出力ファイル名を設定します。	
FILEPLSN	≪省略不可≫
GetPlsN 機能の出力ファイル名を設定します。	

<table>
<tr><th>環境変数名／設定可能な値の範囲と意味</th><th>省略時初期値</th></tr>
<tr><td>FILEPLSP</td><td>≪省略不可≫</td></tr>
<tr><td colspan="2">GetPlsP 機能の出力ファイル名を設定します。</td></tr>
<tr><td>FILESIZE</td><td>0</td></tr>
<tr><td colspan="2">0: データ・ファイル分割をしません。

1 ≦ FILESIZE ≦ 符号無し 64 ビット整数の最大値:

FILESIZE で指定したバイト数を超えないようにデータ・ファイルを複数のピースに分割します。OS のファイルシステムに最大ファイルサイズの制限がある場合に使用します。分割されたピースは下記の書式によりそれぞれが識別できます。

＜テーブル名＞_999999.＜環境変数 DATAEXT（77 頁）＞

999999 は左側0詰6桁の自然数

環境変数 DISCRETE（77 頁）に任意文字列が指定されている場合で、かつに限り、FILESIZE の指定が意味を持ちます。出荷時設定の 2000M (2000 メガバイト) は SQL*Loader へ入力可能なピースの最大サイズよりやや少な目のサイズです。

因みに SQL*Loader へ入力可能なピースの最大サイズは 2 の 31 乗-1 バイトです。

1バルク分の操作で出力されるデータ量よりピースのサイズを小さく出来ません。1バルク分の操作で出力される行数は BULK_SIZE（75 頁） へ指定します。

並列処理により各バルクの出力される順序は一定でないため、分割されたおのおののピースのサイズは実行の都度、変わることがあります（但しファイルサイズは一定です）。非同期ファイル操作：IO_OVERLAP_SCALE（84 頁） を指定している時は、ピースのサイズがばらつくことがあります。

最初のデータファイルに限り、＜テーブル名＞_999999.＜環境変数 DATAEXT（77 頁）＞のファイル書式名に変更される時期が、全てのピースの出力が完了するまで遅延しますので予めご了承ください。

補助単位として k,M,G,T,P を数値の後に追加して使うこともできます。

環境変数 STREAM_LOCATOR（98 頁）に値が設定されている場合、FILESIZE は 0 が指定されているものとみなします。

FILESIZE によるデータファイル分割の影響を受けるファイルの種類はプライマリ・データ・ファイル(*.dat)のみです。

その他の*.clo, *.blo といった LOBFILE 方式（156 頁）や SDF 方式（158 頁）のファイルでは FILESIZE の設定に関わらず、ファイルは分割されません。</td></tr>
<tr><td>FILETABLE</td><td>≪無≫</td></tr>
<tr><td colspan="2">この変数に設定されたファイルに含まれるテーブル名を処理対象とします。LISTTABLE（84 頁）で実在するテーブル名が1つも選択されていない場合には、このファイルに含まれているテーブル名が唯一の選択リストとなります。

ワイルドカード文字を使い複数のテーブルを指定することも可能です。ファイルの位置は INPUT（83 頁）で指定します。

このファイルにはテーブル名の他、任意のコメント文字列を含めることが出来ます。コメント文字列は必ず「#」ではじめてください。コメント文字列は行頭から開始することも行の半ばから開始することも出来ます。</td></tr>
<tr><td>FILETBLN</td><td>≪省略不可≫</td></tr>
<tr><td colspan="2">GetTblN 機能の出力ファイル名を設定します。</td></tr>
</table>

<table>
<tr><th>環境変数名／設定可能な値の範囲と意味</th><th>省略時初期値</th></tr>
<tr><td>FILETBLP</td><td>≪省略不可≫</td></tr>
<tr><td colspan="2">GetTblP 機能の出力ファイル名を設定します。</td></tr>
<tr><td>FILETRACE</td><td>説明部分参照</td></tr>
<tr><td colspan="2">任意文字列：トレースログファイルの名前を任意に与えることが出来ます。ファイル名より前方のパスを省略した時の出力先は環境変数 OUTPUT（89 頁）の指し示す先です。省略時初期値は下記の通りです。トレースログファイルの出力を止める事はできません。
"_mtu"+"_(半角数字 5 桁のプロセス ID)"+"(半角数字 2 桁の連続番号)"+".log"</td></tr>
<tr><td>FINDSTRCMD</td><td>≪無≫</td></tr>
<tr><td colspan="2">Unload/Query 機能の管理作業用スクリプト内で使用される文字列探索プログラムを明示的に指定出来ます。grep (UNIX 用) や findstr (WindowsNT/2000/XP) が使用できない Windows95/98/Me でデータのリロード作業を行う場合に指定して下さい。
例：set FINDSTRCMD=find /n
値が設定されていない場合、使用される文字列探索プログラムは環境変数 WINTARGET（99 頁）に設定された値に依存して決定されます。WINTARGET に値が指定されている場合 findstr /n が、指定されていない場合は grep –n が管理作業用スクリプトへ出力されます。</td></tr>
<tr><td>FPNUMFMT</td><td>≪無≫</td></tr>
<tr><td colspan="2">精度指定のない NUMBER、または FLOAT 型を出力する時の数値書式モデルを指定します。
≪無≫: 9.99999999999999999999999999999999999999EEEE
（小数点の後に 9 が 38 個続く）
任意文字列: 指定した文字列がそのまま数値書式モデルとなります。Ver.2.11 以下のバージョンと等価的な表記は 999999999999999999999999999999999999999 （9 が 39 個続く）です。
NUMBER(10,2) のように精度指定が有れば FPNUMFMT の指定に関係なく最大精度を表現するのに適した数値書式モデルが自動的に選択されます。</td></tr>
<tr><td>GETEXTTBL</td><td>≪無≫</td></tr>
<tr><td colspan="2"><u>≪無≫:</u> 外部表を定義するスクリプトを出力しません。
<u>任意文字列:</u> 出力します。ただし DISCRETE（77 頁）にも任意の文字列が設定されている場合に限ります。出力されたスクリプトを使って外部表を定義し、問合せを行う前に環境変数 DIRALIAS（77 頁）へ指定したディレクトリを指し示すディレクトリ・オブジェクトが事前に作られ、かつオブジェクト権限が問い合わせるユーザへ与えられていることを確実にして下さい。オブジェクト権限は read と write が必要です。</td></tr>
<tr><td>INPUT</td><td>.（カレント）</td></tr>
<tr><td colspan="2">ファイル入力元のディレクトリを指定します。
下記の環境変数が指示している、入力に使うファイル名の置き場所を指示します。
<table><tr><td>環境変数名</td><td>頁</td></tr></table></td></tr>
</table>

環境変数名／設定可能な値の範囲と意味	省略時初期値
<table><tr><td></td><td></td></tr><tr><td>FILEBIND</td><td>80</td></tr><tr><td>FILEEXCPT</td><td>81</td></tr><tr><td>FILEFIXED</td><td>81</td></tr><tr><td>FILETABLE</td><td>82</td></tr><tr><td>POST_REP_EXEC_PLS</td><td>90</td></tr><tr><td>PRE_REP_EXEC_PLS</td><td>91</td></tr></table>	
IO_OVERLAP_SCALE	1
0 を指定すると同期ファイル操作を行います。 1 以上の値を指定すると非同期ファイル操作を行います。 数値は各スレッドに割当てるI/O バッファ数を示します。負数を明示的に指定した時は 0 を指定したのと同じ効果があります。環境変数 PARALLELISM（90 頁）に 2 以上を設定している時は 1～2 が最適です。PARALLELISM に 1 を設定している時は 10～20 が最適です。	
LISTEXCPT	≪無≫
≪無≫: 出力対象表から除外する表は無いとみなします。 テーブル名、ワイルドカード文字を含む文字列、およびこれらを複数個カンマで結びつけた文字列: 指定した文字列にマッチするテーブル名は LISTTABLE にマッチするテーブルから除外します。大文字と小文字は区別されます。通常は大文字で記述して下さい。 ディクショナリ上の表名が小文字で管理されている場合には小文字で記述して下さい。OS の制限を受けなければ最大で 32767 文字まで記述できます。 カンマで区切られた一つひとつの要素は 30 バイト以下の文字列を指定することが出来ます。 ※ワイルドカード文字「%」を指定する場合は前方にもう一つ「%」を付加してエスケープします。下記「注意事項」参照。	
LISTFIXED	≪無≫
LISTTABLE（84 頁と FILETABLE（82 頁）に指定したテーブルのうち、固定長データ形式で出力させたいもののカンマ区切りリストを指定します。 カンマで区切られた一つひとつの要素は 30 バイト以下の文字列を指定することが出来ます。	
LISTTABLE	%
テーブル名、ワイルドカード文字を含む文字列、およびこれらを複数個カンマで結びつけた文字列: 指定した文字列にマッチするテーブル名を出力対象表とします。大文字と小文字は区別されます。通常は大文字で記述して下さい。 ディクショナリ上の表名が小文字で管理されている場合には小文字で記述して下さい。 カンマで区切られた一つひとつの要素は 30 バイト以下の文字列を指定することが出来ます。 ※表名に「_」文字が含まれる場合は「¥」文字でエスケープする	

環境変数名／設定可能な値の範囲と意味	省略時初期値
※ワイルドカード文字「%」を指定する場合は前方にもう一つ「%」を付加してエスケープします。下記「注意事項」参照。	
LOADMETHOD	≪省略不可≫
INSERT: ロード先の空の表に行を追加します。 APPEND: 新しい行を表に追加します。 REPLACE: 表の行を置き換えます。（DELETE を使う） TRUNCATE: 表の行を置き換えます。（TRUNCATE を使う） SQL*Loader の制御ファイルへ含めるロード方法に関するオプションを指定します。大文字、小文字を区別しません。INSERT オプションを使用した場合に限り、管理作業用スクリプトとして更にもう一つ _trunc.sql が OUTPUT によって示される位置へ出力されます。このスクリプトには出力対象表を一括して切り捨てる(TRUNCATE TABLE)為の無名 PL/SQL ブロックが含まれています。このスクリプトを実行するユーザへ ALTER TABLESPACE システム権限が付与されている場合に限り、全ての出力対象表を切り捨てた後に ALTER TABLESPACE 表領域名 COALESCE が実行されます。この、結果切り捨てたセグメントの追加エクステントに占められていた領域の後に生じる断片化した隣接する使用可能エクステントが結合され、より大きなサイズの使用可能エクステントを得ることができます。	
LONGTRANSIT	0
0: LONG 型/LONG RAW 型を CSV/固定長データ・ファイルへ出力します。（無変換） 1: CLOB 型/BLOB 型としてロードできる OS ファイルを作成します。 2: BFILE 型としてロードできる OS ファイルを作成します。 LONGTRANSIT=2 の時、DIRALIAS（77 頁）は必須です。	
MAXLONGSIZE	≪省略不可≫
0 ＜ MAXLONGSIZE ≦ 符号付き32ビット整数の最大値 LONG 型/LONG RAW 型最大バイト数 アンロード対象表に含まれる LONG/LONG RAW 型データの最大長を指定します。この値を超える長さの LONG/LONG RAW データをフェッチすると、「Warning - 戻された LONG 型/LONG RAW 型値はホスト変数の最大長を超えたため切捨てられました。」というメッセージが表示され実際に設定された値のバイト数を超えるバイト列は切り捨てられます。 ※大きな数値をとればその分だけ大きなサイズの LONG 型/LONG RAW 型データを取得する為のメモリを用意することが出来ますが、この値と環境変数 BULK_SIZE（75 頁）を乗じたバイト数のメモリが動的に取得される為過剰に大きくならないように調整する必要があります。	
MERGE_LOBS_INTO_SDF	
≪無≫: 表に内部 LOB 型(CLOB/NCLOB/BLOB)が含まれる場合、LOB 値毎にファイルを作成する動的な LOBFILE 形式で出力します。 任意文字列: LOB 列ごとにひとつのファイル（SDF）を作成し、LOB 値をプライマリ・データ・ファイルへ出力した行の順序と同じ順序で出力します。 LOB 出力に関する詳しい記事は「23.LOB 出力機能について（156 頁）」を参照してください。	
METADIV	≪無≫

環境変数名／設定可能な値の範囲と意味	省略時初期値
≪無≫: GetMeta 機能の実行時にオブジェクトの定義文が環境変数 OUTPUT(89 頁)と FILEMETA(81 頁)に指定したファイルへ全て出力されます。 任意文字列: GetMeta 機能の実行時にオブジェクトの定義文がオブジェクト名+".sql"と言う名前の別ファイルへ出力されます。環境変数 OUTPUT と FILEMETA に指定したファイルには SQL*Plus の @@コマンドを使ってオブジェクト名+".sql"のファイルが呼び出せるように出力されます。オブジェクトがパッケージ(またはタイプ)本体 の場合に限りファイル名がオブジェクト名+"_BODY"+".sql"となります。 リフレッシュグループを定義する無名 PL/SQL ブロックが格納されたファイル名は DBMS_REFRESH_MAKE.sql となります。 ジョブを定義する無名 PL/SQL ブロックが格納されたファイル名は DBMS_JOB_SUBMIT.sql となります。	
MIN_BULK_SIZE	1
1 ≦ MAXLONGSIZE ≦ 符号なし32ビット整数の最大値 バルク数の最小限数を指定します。行サイズの大きな表のアンロードでバルク数が小さくなりすぎる場合にこの値に設定されたバルク数を下回らないようにすることが出来ます。	
MINIDUMP_TYPE	0x0
ハングやクラッシュ時におけるプロセス・ダンプ出力の際のコンテンツを 21 ビット分のフラグ使って指定します。 省略時初期値の 0x0 は MiniDumpNormal「Include just the information necessary to capture stack traces for all existing threads in a process.」です。 詳しくは、下記 URL にあるマイクロソフト社の資料参照。 http://msdn.microsoft.com/ja-jp/library/windows/desktop/ms680519(v=vs.85).aspx	
MINIMUM_FREE_SIZE	0
0: 領域サイズの監視を行いません。出力先がデータで一杯になるまで書き込み続けます。 1 ≦ MINIMUM_FREE_SIZE ≦ 符号なし 64 ビット整数の最大値 出力先ディスクの空き領域チェックを行う時の領域サイズを指定します。指定した領域サイズを空き領域が下回った時、事前に指定しておいた方針に基づいて障害回避の動作を試みます。 補助単位として k,M,G,T,P を数値の後に追加して使うこともできます。 詳しい内容については「18.領域あふれ時のフェイルオーバー(139 頁)」の記事を参照してください。	
MNGSCROUT	0x0000
UnLoad 時の管理作業用スクリプト出力時の動作に作用します。16ビットの整数をC言語の16進数表現で設定します。下記に示す各ビットがマークされていると対応する管理作業用スクリプトが実行されます。	

マークするビット	有効化するスクリプトの種類
16ビット	参照制約
15ビット	検査制約
14ビット	主キー制約・一意キー制約

環境変数名／設定可能な値の範囲と意味	省略時初期値
13ビット / Not NULL 制約 12ビット / データベース・トリガー 11ビット / 索引 10 ～ 1ビット / 未使用（動作上の意味はありません	
DB_FILE_MULTIBLOCK_READ_COUNT	≪無≫
マルチブロック読み取り一回当たりのフロック数について接続時のデフォルト値を上書することが出来ます。	
NMENCLS	≪無≫
≪無≫: オブジェクト名を二重引用符で囲みません。 任意文字列: オブジェクト名を二重引用符で囲み、「引用識別子」と呼ばれる表現での出力をサポートします。マルチバイト文字アルファベット小文字や半角空白を含む特殊なオブジェクト名を表現する場合にこのモードにしますす。 環境変数 NMENCLS に値が設定されている時は、環境変数 SRC_USER（97 頁）, ALTER_DEST（74 頁） に与られたスキーマ名は「引用識別子」と呼ばれる表現で扱われますのでスキーマ名のアルファベット大文字／小文字が区別されるようになります。環境変数 NMENCLS に値が設定されていない時は引用識別子として扱われなくなるので、ALTER_DEST に与えられたスキーマ名は、小文字を含んでいたとしても大文字に変換した上で扱われます。 30 バイト以下の文字列をセットする事ができます。 引用識別子に関する説明は下記の文献を参照してください。 Oracle 製品マニュアル「SQL リファレンス」 2 章 Oracle SQL の基本要素 スキーマ・オブジェクト名および修飾子 スキーマ・オブジェクトのネーミング規則	
NOBYTEORDERMARK	≪無≫
≪無≫: 環境変数 CHARSETID（75 頁）へ UTF-8 系のキャラクタセットが指定された時にデータファイルの先頭へ BOM を出力します。また SDF 方式の LOB 出力時にもファイルの先頭に BOM を出力します。 任意文字列: BOM を出力しません。制御ファイルのマルチバイト文字系のカラムに対して BYTEORDERMARK NOCHECK というオプションが追加されます。	
NUM_ASYNCLOBS	0
≪無≫: :が指定されたものとみなします。 0: LOBFILE 出力を逐次処理します。現在処理中の LOBFILE 出力が終わるまで次の LOBFILE 出力は開始されません。 1≦ NUM_ROWS ≦ 符号なし32ビット整数の最大値： 指定した数の並列度で LOBFILE 出力を並列処理します。 LOB のサイズが数メガバイトを超えるような場合、2 以上の並列度を選択すると性能の向上を期待できます。反対に小さな LOB が沢山ある場合は、ファイルのオープン・クローズに掛かるオーバーヘッドが大きいため並列度を上げて	

環境変数名／設定可能な値の範囲と意味	省略時初期値
もあまり効果がありません。	
NUM_ROWS	0
0 ≦ NUM_ROWS ≦ 符号なし32ビット整数の最大値 表データ取出し条件の最少行数を指定します。出力対象表の行数を数え、この値以上の行を含む表だけを出力することが出来ます。 0 を指定した場合は、行が1つも含まれていない表であっても 0 行分のデータファイルを出力します。 1 を指定した場合は、行が含まれている表だけを出力対象として選びます。	
NUMBER2BCD	≪無≫
7: COMP-3（パック 10 進数） 91: DISPLAY SIGN LEADING SEPARATE 7、91 のいずれかを指定することができます。NUMBER 型の数値を出力する際に上記いずれかの BCD に変換することができます。 なにも指定しなかった場合は、通常通り可変長 10 進数の数値文字列を出力します。	
NUMDAYS_AUDIT	≪無≫
≪無≫: 監査証跡を出力しません 0 ≦ FILESIZE ≦ 符号あり32ビット整数の最大値: 当日から指定した日数を遡った範囲内に作成された監査証跡を出力します。0 は当日分の監査証跡を出力するという意味になります。meta.sql 上の監査証跡はカンマに区切られた8つのフィールドから構成され、左から順に次の意味を示します。 ● 監査証跡が作られた日付時刻 ● 各 Oracle セッションの数値 ID ● セッションの各監査証跡エントリの数値 ID ● 監査対象の操作 ● 操作の影響を受けたオブジェクトの名前 ● OS 上のログイン・ユーザ名 ● OS 上のクライアント識別 ● 操作を行った Oracle アカウント ● 操作の結果(0 以外は異常終了） ● 操作の実行に使用されたシステム権限 監査証跡（SYS.AUD$表）に非常に多くの行が含まれている場合 `オブジェクト監査証跡 : Executing.` という表示のまま長時間待機する場合が有ります。	

<table>
<tr><th>環境変数名／設定可能な値の範囲と意味</th><th>省略時初期値</th></tr>
<tr><td>OPTIMIZER_HINTS</td><td>≪無≫</td></tr>
<tr><td colspan="2">UnLoad 時に指定した文字列をオプティマイザヒントとして、問合せへ与える事ができます。
例えば PARALLEL ヒントを与えて間接的にバッファ・キャッシュを経由せず、ダイレクト・パス・リードをさせたいときなどに使用することができます。
指定するヒントにはエリアスとして ″T1″ という文字列が使用できます。たとえば OPTIMIZER_HINTS へ
PARALLEL(T1,4)
のようなヒントを与えると各テーブルに対して多重度4でパラレル問合せを試みることができるようになります。</td></tr>
<tr><td>OPTIMIZER_MODE</td><td>≪無≫</td></tr>
<tr><td colspan="2">指定した文字列を使って内部的に下記 SQL 文が実行され、デフォルトの最適化アプローチを上書きすることが出来ます。
alter session set optimizer_mode = 文字列
Oracle10g の場合下記の何れかが設定可能です。
first_rows_1, first_rows_10, first_rows_100, first_rows_1000, first_rows, all_rows</td></tr>
<tr><td>OUTPUT</td><td>.(カレント)</td></tr>
<tr><td colspan="2">製品が出力するファイルの格納先のディレクトリ名を指定します。ローカルディスクのフォルダの他にも UNC 表記の共有フォルダを指定することができます。C:¥Documents and Settings¥... といった途中に空白が含まれるディレクトリ名も指定することが出来ます。この時、二重引用符でディレクトリ名を囲む必要はありません。
UnLoad/Query では SQL*Loader 制御ファイルやデータ・ファイルが、GetMeta では meta.sql ファイルが出力される先となります。UnLoad や Query のような I/O 集中型の機能を使う場合は、十分な書込性能を期待できるディレクトリを選んで下さい。ネットワーク上のリモートドライブのディレクトリや、自動的にファイルが圧縮される設定のディレクトリは避けてください。</td></tr>
<tr><td>OVERLAP_BUFFER_LENGTH</td><td>512k</td></tr>
<tr><td colspan="2">出力先ディスクドライブのセクタサイズ≦OVERLAP_BUFFER_LENGTH≦256M の範囲内の整数。この範囲以外の値が指定された場合、省略時初期値が指定されたものとみなします。
補助単位として k,M,G,T,P を数値の後に追加して使うこともできます。
非同期 I/O 時に使用するバッファのサイズを指定できます。殆どの場合 出荷時初期値の 1M バイトで良好なパフォーマンスを得る事ができますが、デバイスとの相性により 1/8 倍～4 倍程度の範囲で増減させるとパフォーマンスが若干向上する場合があります。長過ぎるバッファを使用すると同期 I/O と特性が近似するため、非同期の利点を活かす事ができなくなる傾向があります。</td></tr>
<tr><td>PARTITIONING</td><td>0</td></tr>
</table>

環境変数名／設定可能な値の範囲と意味	省略時初期値

0, 1, 2, 3 のいずれかの値を設定できます。指定した値が 32-bit 整数として扱われますが実際に使用されるのは下位 2-bit 分です。下位 1-bit 目、下位 2-bit 目にはそれぞれ次の意味づけがあります。

設定値	32〜3ビット目	2ビット目	1ビット目
4 以上	無効	−	−
3 (=任意文字列)	未使用	1	1
2	未使用	1	0
1	未使用	0	1
0 (=空文字列)	未使用	0	0
ヒープ構成表の並列アンロード			
パーティション表の並列アンロード、及び集合化クエリ			

● ビットの値が1の場合、その並列処理の結果出力されるデータファイルは分割されます。

分割数はパーティション表の場合パーティションの数と等しく、ヒープ構成表の場合は PARALLELISM1（90 頁）か定数 16 のいずれか小さい数、集合化クエリの場合は要素クエリの数になります。

分割された後のファイル名はパーティション表の場合は半角アンダーバー文字＋パーティション名が、ヒープ構成表と集合化クエリの場合は拡張子を除くファイル名の部分に半角アンダーバー文字＋2 桁左側ゼロ詰めの数値文字列（例：”_01”）が追加されます。

制御ファイル中の LOAD INFILE 句のデータ・ファイル名部分も分割後のファイル名に一致するようプロセスの終盤に書き換えられます。

● 反対にビットの値が 0 の場合、並列処理の結果であってもデータファイルは1つにまとめられます(単一ファイル出力)。

この機能を使用する制限として、I/O が1箇所に集中するため、分割する時より数%程度のスローダウンがあります。また、複数のスレッドがそれぞれ任意のタイミングでデータを書き込むので、結果のデータセットは関係代数的に同一ですが行の順番は実行毎に異なる場合がありますので予めご了承ください。

PARALLELISM	1

0 以上の32ビット正数、複数のスレッドを使用したときの並列度を指定します。0 または負数を明示的に指定した時は 1 を指定したものとみなします。コンピュータに搭載されている CPU 数(コア数)か、それよりも 1 つか2つ多い数が適正値です。過剰な値を指定するとメモリを浪費し、却ってスローダウンすることがあります。LOB 列を含む表の Unload か Query の場合には 2 以上を指定してください。

POST_REP_EXEC_PLS	≪無≫

指定した無名 PL/SQL ブロックを接続中の全てのセッションで実行します。実行する時期は全てのテーブルや任意問合せのデータを読み終えた後です。

EXECUTE …; か、あるいは[DECLARE|BEGIN] … END;で囲まれた、無名 PL/SQL ブロックを指定することが出来ます。最後の文字が半角セミコロンで終了していると無名 PL/SQL ブロックであると仮定して解釈します。最後の文字が半角セミコロンでない場合は、無名 PL/SQL が含まれているスクリプトファイルの名前であると仮定して解釈し、ファイル

環境変数名／設定可能な値の範囲と意味	省略時初期値

が環境変数 INPUT（83 頁）の指し示す先にある場合、それをオープンして文を読取ります。

PRE_REP_EXEC_PLS	≪無≫

PL/SQL を実行する時期が全てのデータの読取前である、と言う点を除いて「POST_REP_EXEC_PLS（90 頁）」で説明した内容と同一です。データの読取前に実行されるので、Oracle Virtual Private Database (VPD)を構成しているシステムからデータを取得するため、アプリケーション・コンテキストを設定する処理の設定に使うことが出来ます。

認可されているアプリケーション・コンテキストを指定しないと、行が戻されないようなセキュリティー・ポリシーが施行されている FGAC 対象表を Unload する場合には、NUM_ROWS（88 頁）を 0 にしてお使い下さい。NUM_ROWS>0 を指定してしまうと、アプリケーション・コンテキストを実行する前に

```
select count(*) from FGAC 対象表
```

と言う問合せを実行している為、1行もカウントされず、結果的にアンロードされない現象になります。Query では特に NUM_ROWS の設定の影響が結果に及ぶ事はありません。

PRINTCOLID	≪無≫

何か文字列が設定されている時に列番号、パーティション位置を含むコメント・ブロック（/* <左側0詰め5桁整数>*/）を出力します。

PRIORITY_CLASS	NORMAL

MTU と MPX 実行時の優先順位クラスを設定します。設定可能な値について順位の低い順に説明しています。アルファベットの大文字小文字は区別されますので全て大文字で入力してください。無効な設定値を設定した時はエラーとはならず、NORMAL が指定されたものとみなします。

MTU と競合するプロセスが他に動作していてそちらの優先度を確保したい場合には NORMAL 未満を選択してください。事実上 HIGH 以上の値を設定して得られる有益な効果は殆どありません。

設定値	説明
IDLE	システムがアイドル状態のときにのみ、このプロセスを実行することを示します。
BELOW_NORMAL	IDLE より高く、NORMAL より低い優先順位を割り当てます。
NORMAL	特別なスケジューリングを必要としない、一般的なプロセスであることを示します。
ABOVE_NORMAL	NORMAL より高く、HIGH より低い優先順位を割り当てます。
HIGH	タイムクリティカルなタスクを実行することを示します。
REALTIME	最も高い優先順位クラスを持つプロセスであることを示します。

QUERYFIX	≪無≫

環境変数名／設定可能な値の範囲と意味	省略時初期値

ここに格納された問合せ結果は固定長データ形式で出力されます。

次のいずれか1つを設定することができます。

1. 任意問合せが格納されたファイルが0個以上含まれるディレクトリ

   ディレクトリ名が指定された場合はそのディレクトリ下に存在する、環境変数 EXTNAMESQL(80 頁)に指定された文字列の拡張子のファイルを、問合せの含まれているスクリプトと仮定して全て実行します。

2. 任意問合せが格納されたファイル

   ファイル名が指定された場合はそのファイルを1つだけ実行します。ディレクトリが明示されなかった場合はそのファイルがカレントディレクトリにあるものと仮定します。

3. 末尾に半角セミコロン1文字が追加された SELECT 文

   SELECT 文の場合はその文だけを実行します。末尾に半角セミコロン1文字があると直接 SELECT 文が与えられたものと解釈します。

どれが指定されたかは値を解析した上で MTU が判断します。この環境変数を指定しない場合は固定長データ形式での出力を要求するクエリが無いものとみなします。

- 1, 2 について補足

集合化クエリと呼ばれる特別なスクリプトファイルの記述法に基づく最適化機能を利用できます。詳しい記事は 21.集合化クエリ(152 頁)を確認してください。

- 3 について補足

SELECT 文の最初の FROM 句が開始される直前に INTO <問合せ別名>という表現の句を含めると問合せの出力を問合せ別名で指定した名前のデータ・ファイルへ保存する事が出来ます。INTO 句を指定しないと、INTO QUERYFIX が指定されたものとみなします。問合せ別名に指定可能な最大文字列長は 30 バイトまでです。使用可能な文字については Windows のファイル名に使用できる識別子と同じ制限を受けます。

SQL に使用できる文字にはパイプ記号、不等号記号、パーセント記号など一部、Windows のコマンド・インタプリタの正常な動作へ影響を与えるものがありますのでそのような文字を含む問合せを使用する場合には二重引用符で問合せ全体を囲んでください。

このとき文字列末端のセミコロンは引用符の内側に配置されるようにして下さい。問合せ文字列のもっとも外側にある二重引用符以外の二重引用符は SQL の一部として解釈されます。

下記にこの機能の有効な使い方の事例を示します。

```
set QUERYFIX="select "ENAME" || ':' || "JOB" as concat_name from "EMP";"
```

問合せの名前は環境変数 QUERYVAR(92 頁)および QUERYFIX(91 頁)の双方を併せて一意になるように与えてください。一意でないものは、特にエラーにはなりませんが後に実行されたものの内容で先に実行されたものが上書きされてしまうのでご注意下さい。

- 全体の補足

環境変数 PARALLELISM(90 頁)が 2 以上の場合、問合せの実行順序は問合せの名前の文字コード順になります。データ量にばらつきのある問合せを同時に実行する場合には、データ量の多い問合せほど文字コード順の前方に配置されるよう問合せの名前を決めると、データ量の多い問合せをより早く開始させる事が出来るので並列処理の効果を受けやすくなります。

QUERYVAR	≪無≫

環境変数名／設定可能な値の範囲と意味	省略時初期値
ここに指定された内容は（CSV,TSV 等の）可変長データを要求するためのものである点を除いて環境変数 QUERYFIX(91頁）で説明した内容と同一です。 また SELECT 文の指定で INTO 句を省略した時のデフォルトは QUERYVAR です。	
QUERYFILENAME	*
ファイル名のベース部分を指定すると、指定した名前にマッチする任意問合せファイルだけを限定して扱います。ベース部分には、Windows のコマンドに使用できるワイルドカード文字を含めることが可能です。なにも名前を指定しないと半角アスタリスク1文字が指定されたものとみなします。	
RECLENGTH	≪無≫
≪無≫: CSV データ・ファイルの各レコード先端部に 10 桁のレコード長を示す文字列を付加しまません。レコード間の区切りを改行文字の出現によって検知します。 任意文字列:レコード長文字列を付加します。レコード間の区切りをレコード先端部に示された 10 桁のレコード長で検知します。文字列を指定すると制御ファイルの LOAD INFILE 句に"VAR"オプションが追加されます。"VAR"オプションに対応するデータ・ファイルの場合、非表示文字や改行文字を含むデータであってもSQL*Loader が正しく取り込める利点があります。また改行文字を探す処理が省かれるので数 10%程度の所要時間削減効果が得られます。ただし Oracle8 の SQL*Loader ではでは VAR オプションを使用することが出来ませんのでご注意ください。固定長データ形式のデータ・ファイル出力では特定の意味を持たないので 10 桁のレコード長は出力されません。	
RECOVERABLE	≪無≫
≪無≫: ダイレクト・パス・ロード（環境変数 USUALPATH (99頁)に何も指定しない時）で UNRECOVERABLE オプションを使用します。制御ファイルの LOAD 句の前に"UNRECOVERABLE"オプションが記述されます。 任意文字列: 常に RECOVERABLE オプションを使用します。制御ファイルの表記上 RECOVERABLE がデフォルトなので LOAD 句の前にはなにも記述されません。 ※UNRECOVERABLE オプションはダイレクト・パス・ロード時にだけ使用することが出来ます（SQL*Loader の仕様）。データベースがアーカイブログモードで動作している時に所要時間の短縮効果が得られます。このオプションを使用してデータをロードした時は直ちにバックアップを取得して下さい。バックアップを取得する前に媒体障害が発生するとロードしたデータを回復できない為です。高速化と引替えに受け取るリスクについて「16. ダイレクト・パス・ロード（132頁）」に説明しましたのでご参照ください。	
RERUNPOINT	≪無≫
≪無≫:最初から実行します。 表名を表す文字列: 再実行を行う場合に中断個所の表名を指定します。表名に半角アルファベット文字が含まれている場合、大文字と小文字を区別します。通常は大文字を指定して下さい。 実在しない表名を指定してもエラーにはならず、何も指定されなかったのと同じように動作（最初から実行）します。	
REVERSEBYTEORDER	≪無≫
≪無≫:バイト順序を変えません。Windows のベースである Little Endian を選択します。 任意文字列: バイト順序を Big Endian へ変更します。 任意の文字列を指定すると CHARSETFORM へ 2 が設定されている時に出力される NCHAR 型/ NVARCHAR2 型由	

環境変数名／設定可能な値の範囲と意味	省略時初期値
来の UTF-16 文字のバイト順序を逆順にすることが出来ます。 これは Intel 系プロセッサで多バイト長文字や多バイト長整数を表現する場合のバイト順序が Little Endian であるのに対し、IBM 系のメインフレーム/汎用機ではバイト順序が Big Endian であることに起因する非互換性を解決する為のものです。 この指定を行った場合に 出力される UTF-16 BE 文字は通常のテキストエディタでは正常に表示することが出来ないのでご留意下さい。	
ROWID_SPLIT_MIN_SIZE	0
0: ヒープ構成表の並列アンロードをしません。 1≦ROWID_SPLIT_MIN_SIZE≦符号付 64 ビット整数の最大値:指定した数のバイト数を超えるヒープ構成表セグメントについて、ROWID ベースの範囲に基づく分割を行い、それぞれの部分を並列にアンロードします。 分割数の決定は次の手順に従います。 単一ファイル出力の時: ROWID_SPLIT_NUM_PARTS(94 頁)に指定した値 非単一ファイル出力の時: PARALLELISM(90 頁)に指定した値 上記の値と定数 16 を比較し、いずれか小さいほうが採用されます。 補助単位として k,M,G,T,P を数値の後に追加して使うこともできます。 SORTING(96 頁)に任意文字列が指定されている場合と、ROWS(94 頁)に任意文字列が指定されていない場合は 0 が指定されたものとみなします。また、対象のヒープ構成表の CBO 統計が見つからない(アナライズされていない)場合は、分割対象外とされます。 ヒープ構成表セグメントの比較するバイト数は、トレースファイルの Order タグで識別される部分のログに表示されます。下記にログのサンプルを示します。 `2013-07-22 14:30:16.497 [1318] [Order]     1:    91M (*****) NON_PART_TABLE` [Order] と表示された箇所の後のフィールドに"91M"という表示があります。これはセグメントのサイズが 91MByte であることを示しています。例えば"NON_PART_TABLE"表を分割対象にしたい場合は「ROWID_SPLIT_MIN_SIZE」へ 91M 以下の数値を指定します。	
ROWID_SPLIT_NUM_PARTS	4
ヒープ構成表の並列アンロードと単一ファイル出力を組み合わせて使用する場合この変数に指定した値の並列度でアンロードを行うことが出来ます。 指定できる並列度の上限は 16 (MAX_PARALLELISM) です。値を設定しない場合と、0 や MAX_PARALLELISM を超える値を指定すると 4 が指定されたものとみなして動作します。	
ROWS	≪無≫
≪無≫: CSV/固定長データを出力しません。環境変数 DISCRETE(77 頁)が指定されていなければ 0 バイトのデータ・ファイルが作成されます。データ・ファイル以外は通常どおり出力されるので大量データのアンロード前にリハーサルをすることが出来ます。 任意文字列: CSV/固定長データを出力します。	

<table>
<tr><th>環境変数名／設定可能な値の範囲と意味</th><th>省略時初期値</th></tr>
<tr><td>ROWSEPARATOR</td><td>0x0d0x0a、または 0x0a</td></tr>
<tr><td colspan="2">行セパレータ用の文字列を指定することができます。表記可能な ASCII 文字の他、”0x”が前置きされた 2 桁の 16 進数の 0 個以上の並びを指定することができます。
“0x00”を指定した場合はゼロ文字が出力されるのではなく行セパレータが出力されないという意味になります。SAM（順編成方式）ファイルを出力する際の設定事項の一部として”0x00”を利用することができます。
SAM ファイル出力の設定方法については「22.メインフレーム向けデータ移行（154 頁）」にある記事を参照してください。
なにも指定しなかった場合の省略時デフォルトは WINTARGET（99 頁）に設定した値に依存します。つまり、WINTARGET へ true などの任意文字列が設定されていれば Windows で多用される 0x0d0x0a（CR+LF）が選択されます。文字列が何もなければ UNIX で多用される 0x0a(LF)が選択されます。</td></tr>
<tr><td>SERVEROUTPUT</td><td>0</td></tr>
<tr><td colspan="2">≪無≫: SERVEROUTPUT をしません。
1 以上の32ビット正数：MTU 内部で実行された PL/SQL 内で DBMS_OUTPUT パッケージを使って出力された診断情報を画面へ表示することが出来ます。数値は DBMS_OUTPUT に準備されるバッファの量です。</td></tr>
<tr><td>SHOWCLTDTM</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫: クライアントの日付時刻をレポートしません。
任意文字列: クライアントの日付時刻をローカル・タイムゾーン付でレポートします。レポートするタイミングは Oracle サーバへ接続した時と、接続を開放した時の2回です。</td></tr>
<tr><td>SHOWSTAT</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫:セッション統計情報をレポートしません。
任意文字列: セッション統計情報をレポートします。</td></tr>
<tr><td>SHOWSTMT</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫:エラー発生時の診断出力時に SQL や PL/SQL を含めません。
任意文字列: SQL や PL/SQL を含めて出力します。</td></tr>
<tr><td>SHOWVERS</td><td>≪無≫</td></tr>
<tr><td colspan="2">≪無≫:下記の情報を表示しません。
任意文字列: 下記の情報を表示します。製品が期待通りに動作しない場合の診断出力に使用します。
● プログラムファイル名の場所
● ロードされた OCI.DLL の場所
● OCI のバージョン
● 接続先 Oracle サーバのバージョン
表示例：</td></tr>
</table>

環境変数名／設定可能な値の範囲と意味	省略時初期値
プロセスの位置は C:¥Program Files¥PLUMSIX¥mtu_4.2¥mtu.exe です。 ＯＣＩの位置は C:¥oracle¥product¥11.2.0¥client_1¥bin¥OCI.DLL です。 … ＯＣＩのバージョンは 1102000100 です。 サーバのバージョンは 1002000300 です。	
SKIPSCR	
≪無≫: 管理作業用スクリプトを出力します。UnLoad で出力されるファイルのうち*.dat と*.ctl だけが更新されます。 任意文字列: 管理作業用スクリプトを出力しません。	
SORTING	≪無≫
≪無≫: 行を並び替えません。 任意文字列: 主キーまたは一意キー制約が施行されていればその順序で行を並び替えます。両方があれば主キーが、主キーが無くて複数の一意キーがあれば名前の昇順でもっとも小さなものが選択されます。また並び替えられた表の制御ファイルには SORTED INDEXES オプションが付加されます。ただし、パーティション表に限り、PK や AK が施行されたままデータをロードすると一意索引が無効になる為、このオプション指定を無視します。リロード時に並び替えプロセスが省略される為索引のリビルド処理が最適化され、数%の所要時間削減効果があります。データ量が多く並び替え時にディスク・ソートの発生が避けられないケースでは更にそれ以上の所要時間削減効果が得られます。 主キー制約や一意キー制約を施行する索引のクラスタ化係数が最少化されるのでロード後にこの索引を使う範囲検索のブロック読込数を削減することが出来ます。ただしその他の索引についてはむしろクラスタ化係数が増大することもあるので常に効果があるとは限りません。SORTED INDEXES はダイレクト・パス・ロードが有効な場合（環境変数 USUALPATH (99頁)に何も指定しない時）だけ使用できるオプションです。 その他にはデータ・ファイル同士で diff を取って比較する為に使用することも出来ます。	
SORT_AREA_SIZE	≪無≫
指定した数値文字列を使って内部的に下記 SQL 文が実行され、ソートやジョイン時に使用するワーク用のメモリサイズを調節することが出来ます。 alter session set sort_area_size = 数値文字列 この数値が高いとハッシュ結合とソートマージ結合のコストが下がるためCBOは全表走査を選択する傾向が強くなります。Query 機能使用時に大量の行を戻す結合文を実行する際には、パフォーマンス改善の手段として利用できます。 自動 PGA 管理（pga_aggregate_target > 0）のインスタンスに接続している場合は、この値を指定しても特に効果はありません。	
SQLFMT	0x0000
≪無≫: マテリアライズド・ビュー、ビューの定義問合せ、トリガー定義 PL/SQL、名前付き PL/SQL に字下げ、改行等を適切に含めて読み易くする加工をしません。ディクショナリに格納されているままの状態で出力します。 文の整形加工を行いたい時は16ビットの整数をC言語の16進数表現で設定します。下記に示す各ビットがマークされていると対応する SQL や PL/SQL が 読み下し易い書式に加工されます。 16ビット目 : マテリアライズド・ビューの定義問合せ	

環境変数名／設定可能な値の範囲と意味	省略時初期値

15ビット目：ビューの定義問合せ

14ビット目：データベース・トリガーの定義 PL/SQL ブロック

13ビット目：名前付き PL/SQL ブロック

12 ～ 1ビット目：未使用（動作上の意味はありません）

※現在のバージョンにおいて名前付き PL/SQL ブロックの整形加工は機能的に未熟な為アクセス保護違反等のエラーが発生する場合があります。予めご了承の上お使いください。

SRC_USER	≪無≫

≪無≫: 対象スキーマは第1引数で指定したスキーマと実行時ーザは同一とみなします。

任意文字列: 対象スキーマは文字列で指定したスキーマになります。値を引用識別子として扱うかそうでないかは、環境変数 NMENCLS（87 頁）の設定値に依存しします。下記にその事例を示します。

SRC_USER の値	NMENCLS 設定有 （出荷時初期値）	NMENCLS 設定無
scott	scott	SCOTT
ScOtT	ScOtT	SCOTT
SCOTT	SCOTT	SCOTT

接続先のデータベースに指定したスキーマが実在しない場合はエラーメッセージを表示し異常終了します。また 30 バイトを超えるスキーマ名を与えると異常終了します。

STDOUT	0x0000

データ・ファイル、制御ファイル、および LOB ファイルの出力先は環境変数 OUTPUT（89 頁）へ与えた場所に設定されていますが、この設定を標準出力や環境変数 STREAM_LOCATOR（98 頁）へ指定したあて先へ変更する事が出来ます。それぞれ与えた値の下位 1 ビット目から 3 ビット目に対応します。

設定値	意味
0x0000 または設定無し	データ・ファイル、制御ファイル、LOB ファイル共に環境変数 OUTPUT へ指定した出力先へ出力されます。標準出力は使用しません。
0x0001	データ・ファイルだけ、標準出力へ出力されます。
0x0002	制御ファイルだけ、標準出力へ出力されます。
0x0003	制御ファイル、データ・ファイルの両方を標準出力へ出力します。出力する順番は制御ファイル→データ・ファイルです。
0x0004	LOB 値だけ、標準出力へ出力されます。SDF 方式／LOBFILE 方式のいずれにも対応します。
上記以外	下位 3 ビット分が比較に使用され、残りの上位ビットの値は無視されます。

0 以外に設定して使う場合、エラーメッセージや進捗表示など従来から標準出力を使用している表示要素と区別がつかないので、例えば下記のように env.bat ファイルの環境変数 FILELOG（81 頁）へログ・ファイル名を設定してお使いください。

環境変数名／設定可能な値の範囲と意味	省略時初期値
`set FILELOG=mtu.log` 環境変数 FILELOG に有効なファイル名を指定すると OUTPUT の位置に指定した名前で従来の標準出力への表示要素がファイルとして保存できます。上記の設定により、画面には 画面出力はログ・ファイルへ保存します。<OUTPUT へ指定したディレクトリ名>¥<ファイル名> が表示されますがこのメッセージは標準エラー出力に表示されるので標準出力へ混入することはありません。	
STREAM_LOCATOR	≪無≫
≪無≫: リダイレクトされたストリームを標準出力(stdout)へ出力します。 任意文字列: リダイレクトされたストリームを文字列で指定した場所のストリームへ出力します。 「スキーム名+":// "+ストリームの名前」の書式をもつ文字列を指定します。このバージョンで使用可能なスキーム名には「ipc_pipe」と「named_pipe」があります。 環境変数 FILESIZE(82 頁)は STREAM_LOCATOR が設定されている場合、0 が指定されたものとみなします。 この環境変数の詳しい使い方については「19.STREAM_LOCATOR の使用方法(142 頁)」にある記事を参照してください。	
TAF_DELAY_TIME	60
0≦TAF_DELAY_TIME≦300 フェイルオーバー開始イベントが発生してから、代替先インスタンスの探索を開始するまでの待機時間を秒単位で指定します。上記範囲外の数値が指定された場合は 60 が指定されたものとみなします。	
TERMINATOR	≪省略不可≫
データ・ファイル上の隣り合う列のデータ同士を区切る文字か文字列を指定します。※通常は「,」(半角カンマ文字)を指定します。 0x を前置した 16 進数表現の文字、文字列を使用する事も出来ます。	
TIMESTAMP_MASK	
TIMESTAMP/TIMESTAMP WITH LOCAL TIME ZONE 型の書式マスク文字列を指定します。 接続先オラクルのバージョンが 9i 以上のときに設定が有効です。	
TIMESTAMP_TZ_MASK	YYYYMMDDHH24MISSXFFTZHTZM
TIMESTAMP WITH TIME ZONE 型の書式マスク文字列を指定します。接続先オラクルのバージョンが 9i 以上のときに設定が有効です。	
THREAD_PRIORITY	NORMAL

環境変数名／設定可能な値の範囲と意味	省略時初期値

MTU から生成されるスレッドの相対優先順位値を設定します。MTU の優先順位クラス：PRIORITY_CLASS（91 頁）と、スレッドの相対優先順位値によって、各スレッドの基本優先順位レベルが決まります。下表は設定可能な値について説明しています。アルファベットの大文字小文字は区別されますので全て大文字で入力してください。無効な設定値を設定した時はエラーとはならず、NORMAL が指定されたものとみなします。事実上 HIGHEST 以上の値を設定して得られる有益な効果は殆どありません。

設定値	説明
IDLE	プロセスの優先順位クラスが IDLE、BELOW_NORMAL、NORMAL、ABOVE_NORMAL、HIGH のいずれかである場合は、基本優先順位レベルとして 1 を指定します。プロセスの優先順位クラスが、REALTIME である場合は、基本優先順位レベルとして 16 を指定します。
LOWEST	スレッド標準の相対優先順位値より 2 ポイント低い相対優先順位値を指定します。
BELOW_NORMAL	スレッド標準の相対優先順位値より 1 ポイント低い相対優先順位値を指定します。
NORMAL	スレッド標準の相対優先順位値を指定します。
ABOVE_NORMAL	スレッド標準の相対優先順位値より 1 ポイント高い相対優先順位値を指定します。
HIGHEST	スレッド標準の相対優先順位値より 2 ポイント高い相対優先順位値を指定します。
TIME_CRITICAL	プロセスの優先順位クラスが IDLE、BELOW_NORMAL、NORMAL、ABOVE_NORMAL、HIGH のいずれかである場合は、基本優先順位レベルとして 15 を指定します。プロセスの優先順位クラスが REALTIME である場合は、基本優先順位レベルとして 31 を指定します。

USERID	≪無≫

MTU の動作に影響を与える変数では有りませんが、Env.bat スクリプト中で値が定義され、MTU の起動時引数として GetFkrb.bat、GetMeta.bat、GetPlsN.bat、GetPlsP.bat、GetTblN.bat、GetTblP.bat、Install.bat、QikExp.bat、Query.bat、UnLoad.bat の各スクリプト中で参照されています。8.8.接続先の指定（47 頁）参照

USUALPATH	≪無≫

≪無≫: ダイレクト・パス・ロードが可能な表であればスクリプト出力時にその指定が選択されます。SQL*Loader の起動時オプションの parfile に_par0.txt がセットされます。

任意文字列: 常に従来型パス・ロードが選択されます。即ち parfile に_par1.txt がセットされます。

WINTARGET	≪無≫

≪無≫: SQL*Loader の実行時プラットホームが UNIX。

任意文字列: SQL*Loader の実行時プラットホームが Windows。

CSV データ形式出力でかつ、環境変数 RECLENGTH（93 頁）へ任意文字列が指定されているときに限り、この変数の設定が意味を持ちます。UNIX の場合リターンコードを 1 文字と数え、Windows の場合 2 文字と数えます。例えば1レコード分が100桁のCSVデータである場合、そのレコードの先端部分にレコード長を示す数値文字列10桁が加わり合計110桁になります。このとき、この環境変数に任意文字列が設定されていると数値文字列は「0000000102」を示し任意文字列が設定されていないと数値文字列は「0000000101」を示します。

12. 使用方法

マルチ・スレッド・アンローダー はバッチ処理方式のプログラムです。従いましてプログラムの動作に必要な諸条件は全てプログラム実行前に決定され環境変数ファイル env.bat に記述されていなければなりません。実行の前に環境変数が正しく設定されているかどうか確認して下さい。環境変数の説明については「11. 環境変数について(70 頁)」を参照して下さい。

12.1. 表示項目の共通部分

製品プログラムの全機能は実行中に共通の画面表示を出力します。ここではその出力内容について説明します。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>xxxxxxx
Multi-threaded Unloader version 4.2                                        (a)
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
Windows 7 Professional Service Pack 1 (build 7601), 64-bit native          (b)
Number of NUMA nodes: 1
Number of physical processor packages: 1
Number of processor cores: 2
Number of logical processors: 2
Number of processor L1/L2/L3 caches: 4/1/0
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx を実行します...                              (c)
クライアント・キャラクタセット ID は xxx (xxxxxxxx) です。                    (d)
ネット・サービス名:xxxxxxxxxx へユーザ名:SYSTEM として接続します。
しばらくお待ちください.....
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production   (e)
With the Partitioning, OLAP, Data Mining and Real Application Testing options
接続先は SYSTEM@xxxxx.xxxxxx.CO.JP (ネット・サービス名:xxxxxxxxxx) です。      (f)
サーバー・タイプは xxxxxxxxx です。                                        (g)
監査接続ＩＤは xxxxx です。                                                (h)
サーバーの日付時刻は YYYY-MM-DD HH:MI:SS です。                             (i)
対象スキーマは xxxxx です。                                                (j)
         :
                      ＜実行中の各機能に特有の表示＞
         :
処理対象となった表の総数      :          xx
ラウンドトリップ (回)         :         xxx                                 (k)
出力バイト総数 (kbytes)       :       xx.xx                                 (l)
所要時間 (秒)                 :       xx.xx                                 (m)
処理速度 (kbytes/秒)          :        x.xx                                 (n)
正常終了しました。                                                         (o)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. プログラムの表題がバージョンの符号とともに表示されます。

b. プラット・フォームに関する情報を表示します。前半はオペレーティングシステム、後半はプロセッサに関する情報を示します。

   オペレーティングシステムについては GetNativeSystemInfo[16]という API が戻す値を使用しています。

[16] http://msdn.microsoft.com/en-us/library/windows/desktop/ms724340(v=vs.85).aspx

プロセッサについては GetLogicalProcessorInformation[17]という API が戻す値を使用しています。

c. 各機能と「xxxxxxx を実行します…」の部分に表示される表題の関係について下記に示します。

表 16 各機能と出力される表題

機能	表題
Install	製品コンポーネントのインストール
UnLoad	表データ・ファイル出力
Query	任意問合せファイル出力
GetMeta	スキーマオブジェクト定義 SQL 文出力
ConsDisable	整合性制約の無効化
ConsEnable	整合性制約の有効化
GetFkrb	外部キー再構成スクリプト出力
GetTblP	表間参照階層図（正展開）の出力
GetTblN	表間参照階層図（逆展開）の出力
GetPlsP	名前付き PL/SQL 間参照階層図（正展開）の出力
GetPlsN	名前付き PL/SQL 間参照階層図（逆展開）の出力

d. クライアント・キャラクタセットに関する情報が表示されます。代表的な ID と Oracle キャラクタ・セット名は下記のようなものがあります。

ここに表示される ID は環境変数 CHARSETID（75 頁）で設定内容が反映されます。環境変数 NLS_LANG が指定されている場合は第3セグメントに指定されたキャラクタセットが反映されますが、CHARSETID の同時に指定されている場合は CHARSETID が優先されます。

表 17 Oracle キャラクタ・セット名とキャラクタ・セット ID

ID	Oracle キャラクタ・セット名
873	AL32UTF8
871	UTF8
832	JA16SJIS

e. 接続先オラクルデータベースのバージョンが表示されます。

f. プログラムの第1引数に指定した接続先IDのユーザ名にあたる文字列と、1文字分の「@」記号と global_name ビューの global_name 列から戻される値が連結されて表示されます。「(ネット・サービス名:xxxx)」の部分はプログラムの第1引数にネット・サービス名が指定されたときに限り出力されます。

g. 接続先のサーバー・タイプを表示します。DEDICATED と表示された場合は専用サーバー、SHARED と表示された場合は共有サーバー[18]へ接続されたことを示します。

---

[17] http://msdn.microsoft.com/en-us/library/windows/desktop/ms683194%28v=vs.85%29.aspx

[18] 専用サーバー接続と共有サーバー接続の違いについて一般に言える事は下記の通りです。共有サーバー構成は多数のネットワーク接続から送られてくる短くて到着頻度の高い要求を処理するのに向いています。比較的少数のサーバー・プロセス生成で要求に応えられる為、サーバ・システムに搭載されたメモリに占める総 PGA サイズの割当てを少なくできるため余剰なメモリを他の需要に融通できるメリットがあります。専用サーバ構成は最大の入出力性能を要求される処理に向いています。サーバー・プロセスが接続毎に生成される為、メモリ消費量の変動が著しく共有サーバー構成よりも重点的に監視を続ける必要があります。

h. サーバ･コンピュータから戻される日付時刻が表示されます。

i. 監査接続IDを表示します。この値を使うと接続先データベースの v$session 表からこのプログラムの稼動状況をモニターすることができます。v$session 表は sys ユーザのオブジェクトであるため表示させるためには sys ユーザで接続するか、select any table システム権限を与えられたユーザで接続しなければなりません。問い合わせの例は下記を参考にしてください。

```
例：select * from v$session where audsid = 監査接続ＩＤ;
```

また、audit session （システム権限 create session の行使に関する監査）が設定されている場合、この接続に関する監査証跡が記録されます。問い合わせの例は下記を参考にしてください。

```
例：select * from dba_audit_session where sessionid = 監査接続ＩＤ;
```

ローカル接続のうちパスワードファイル認証以外の認証方式（OS認証、外部認証サーバによる認証）では監査対象外の接続なので 0 が戻されます。

j. 処理対象のスキーマが表示されます。環境変数 SRC_USER（97 頁） に指定した値と同じです。

k. 実行ユーザへ SELECT ANY TABLE システム権限か、あるいは SYS ユーザの V_$SESSTAT 表、V_$SESSION 表、V_$STATNAME 表の全てに対する SELECT 権限が付与されている場合に限り表示されます。表示される数値はサーバー・クライアント間を往復した通信の回数を示しています。環境変数 BULK_SIZE（75 頁）の値を大きくすればするほどラウンドトリップの回数は減り、特に WAN などの低速リンクを介した接続を使用しているときにパフォーマンスの改善が図れます。

l. プログラムがファイル出力したデータ量をキロバイト数で表示します。1kbytes は 1024bytes を示します。端数は小数点第3位以下が切り捨てられます。

m. プログラムがデータベースへ接続してから機能を実行し終わるまでの所要時間を秒で表示します。端数は小数点第3位以下が切り捨てられます。

n. 出力バイト総数を所要時間で除した数値が表示されます。端数は小数点第3位以下が切り捨てられます。この数値はパフォーマンスを定量する為の目安となります。この数値に影響を与える要因として依存度の高いものから挙げると、(1)製品プログラムを実行するコンピュータの性能、(2)ネットワークのデータ転送速度、及び(3)データベースが稼動するコンピュータの性能となります。

o. プログラムがエラー無しで終了した場合、例のとおり表示されます。このときデータベースに対してコミットが実行されます。エラーが発生した場合「異常終了しました。」と表示され、データベースに対してロールバックが実行されます。また、プログラム実行中に CTRL-C の入力が行われユーザから中断が要求された場合、「要求により中断しました。」と表示されロールバックが実行されます。

12.2. UnLoad 機能

1. コマンドプロンプトを表示させ、製品フォルダ（43 頁）にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について（70 頁）」を参照して下さい。

2. Env.bat を保存した後、UnLoad.bat を実行します。

3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>unload
Multi-threaded Unloader version 4.2
                    :                                    :
                    「12.1.表示項目の共通部分(100頁」参照
                    :                                    :
表に含まれている行数の測定      :              5:     0.00
```

```
(a)
複数の接続を確立します。要求された並列度は 4 です。
整合性制約情報-Z                :            19:    0.00
整合性制約情報-B                :             4:    0.28
出力対象の表名                  :             8:    0.39
複数の接続を確立します。要求された並列度は 4 です。
データ・ファイルと制御ファイルを出力中です。output
(b)
TableName                       :       NumRows:  Elapse:Rows/s:Bytes: kB/s:   Len  (c)
DEPT                            :             4:    0.01:   279: 160.:   11:   40.
EMP                             :            12:    0.01:  1070: 869.:   76:   72.
SALGRADE                        :             5:    0.01:   517: 182.:   18:   36.
TEST18                        P2:             2:    0.01:   154: 122.:    9:   61.
TEST1                           :             1:    0.02:    40: 245.:   10:  245.
FK constraints                  :             1:    0.18
PK or AK constraints            :             3:    0.00
セグメント属性                  :           172:    2.45
一意キー制約の構成列名          :             5:    0.00
参照整合性制約の構成列名        :             1:    0.00
関数ベース索引式                :             4:    0.43
サブパーティション化キー列名    :             7:    0.01
パーティション化キー列名        :            12:    0.02
列属性                          :           342:    0.01
索引サブパーティション属性      :            16:    0.02
表サブパーティション属性        :            48:    0.03
索引構成列名                    :            38:    0.02
索引パーティション属性          :            20:    0.05
表パーティション属性            :            28:    0.03
パーティション索引属性          :            10:    0.01
パーティション表属性            :             7:    0.01
表属性                          :            29:    0.02
索引属性                        :             4:    0.01
SQL*Loader のコマンド行作成     :             8
処理対象となった表の総数        :             8                                     (d)
ラウンドトリップ（回）          :           315
出力バイト総数（kbytes）        :         11.19
所要時間（秒）                  :          5.47
処理速度（kbytes/秒）           :          2.04
正常終了しました。                                                                  (e)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. 環境変数 LISTTABLE(84 頁), LISTEXCPT(84 頁), FILETABLE(82 頁), FILEEXCPT(81 頁) へ指定した値に該当するそれぞれの表について行数を調べます。この行数が環境変数 NUM_ROWS(88 頁) へ設定した値に満たない場合、その表をデータ取り出しの対象から外します。

b. データ･ファイルと制御ファイルを出力する先のディレクトリ名が表示されます。このディレクトリ名は環境変数 OUTPUT(89 頁)に指定された値が使われます。

c. データ･ファイルと制御ファイルの出力対象になっている表の以下の属性を示します。

表 18 UnLoad 時に表示される情報

項目名	説明
TableName	表の名前（または環境変数 PARTITIONING(89 頁)が指定されていてかつ表がパーティショニングされているときはパーティション、またはサブ・パーティション名が併せて表示されます）
NumRows	出力した行数
Elapse	所要時間（単位：秒）
Rows/s	1秒あたりの平均出力行数
Bytes	出力バイト数。バイト数の位取りに応じて最も右側の桁が"."，"k"，"M"，"G"，"T"，"P"という表示に代わります。これはそれぞれ 10 の 0 乗、10 の 3 乗、10 の 6 乗、10 の 9 乗、10 の 12 乗、10 の 15 乗を意味しています。
kB/s	1秒あたりの平均出力キロバイト数
Len	1行あたりの平均バイト数

各属性のバイト数には環境変数 STREAM_LOCATOR(98 頁)と ENCLOSER, ENCLOSER2(78 頁)に指定した区切文字(または区切文字列)及び改行文字の文字数も含みます。1秒あたりの平均出力行数の端数は小数点第1位以下が切り捨てられます。1行あたりの平均バイト数は小数点第1位以下が切り捨てられます。この平均バイト数はデータ・ファイルへ出力したときのサイズであって、Oracle 内部に格納されている状態の行のサイズと相関はありますが関数ではありません。従いましてこの値を領域サイズ見積もりに使用するのは適切でないので予めご了承の上お使いください。

環境変数 BULK_SIZE(75 頁) と FEEDBACK(80 頁)を乗じた数の行数が取り出される度に表示が更新されます。

FEEDBACK が 0 の場合、(c)は表示されません。また CTRL-C が入力され、プログラムの中断が要求されると FEEDBACK の値に関係なく「CAN」と表示されます。

d. データ・ファイルと制御ファイルの出力対象になった表の総数が表示されます。

e. 「正常終了しました。」が表示されたら環境変数 OUTPUT(89 頁)へ指定したディレクトリにデータ・ファイルやスクリプトファイルが出力されていることを確認して下さい。以下、実際にデータ・ファイルを出力した例を引用しながら説明します。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>dir output
 ドライブ C のボリューム ラベルは HD01 です
 ボリューム シリアル番号は 1431-E578 です

 C:¥borland¥USER¥mtu のディレクトリ

2008/08/01  10:19    <DIR>          .
2008/08/01  10:19    <DIR>          ..
2008/08/01  10:19               304 DEPT.ctl                              (a)
2008/08/01  10:19               160 DEPT.dat                              (b)
2008/08/01  10:19               697 DEPT_DEPS.sql                         (c)
2008/08/01  10:19               631 EMP.ctl
2008/08/01  10:19               869 EMP.dat
2008/08/01  10:19               651 EMP_DEPS.sql
2008/08/01  10:19               303 SALGRADE.ctl
2008/08/01  10:19               182 SALGRADE.dat
2008/08/01  10:19               318 SALGRADE_DEPS.sql
2008/08/01  10:19               910 TEST1.ctl
2008/08/01  10:19               245 TEST1.dat
2008/08/01  10:19             1,195 TEST18_DEPS.sql
```

```
2008/08/01  10:19               592 TEST18_P1.ctl                         (d)
2008/08/01  10:19                 0 TEST18_P1.dat                         (e)
2008/08/01  10:19               592 TEST18_P2.ctl
2008/08/01  10:19               122 TEST18_P2.dat
2008/08/01  10:19               592 TEST18_P3.ctl
2008/08/01  10:19                 0 TEST18_P3.dat
2008/08/01  10:19               592 TEST18_P4.ctl
2008/08/01  10:19                 0 TEST18_P4.dat
2008/08/01  10:19               312 TEST1_DEPS.sql
2008/08/01  10:19                 0 TEST1_R0000000001_C0006.clo           (f)
2008/08/01  10:19               229 _conn.sql                             (g)
2008/08/01  10:19             1,459 _disable_deps.sql                     (h)
2008/08/01  10:19               729 _make.bat                             (i)
2008/08/01  10:19                65 _par0.txt                             (j)
2008/08/01  10:19               102 _par1.txt                             (k)
2008/08/01  10:19                31 _wait_here.bat                        (l)
              28 個のファイル              11,882 バイト
               2 個のディレクトリ  25,260,097,536 バイトの空き領域

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. 表データに関するファイルは全てファイル名先頭辞として表の名前が与えられています。表の名前を持つファイルのうち、拡張子に「ctl」がついているものは SQL*Loader 用の制御ファイルです。

b. また拡張子に「dat」がついているものはデータ・ファイルです。データ・ファイルの拡張子は環境変数 DATAEXT（77 頁）へ任意の文字列を設定することにより変更することが出来ます。

   データ・ファイルは環境変数 CHARSETID（75 頁）に設定されたキャラクタ・セット ID によってエンコーディングが変化します。詳しくは「13.Unicode 対応について (125頁)」を参照してください。

c. 拡張子が「sql」で、先頭辞に表の名前が与えられているファイルは、表に付随する索引を定義し、整合性制約、データベース・トリガー、およびその表の候補キーを外部キーに持つ表の参照整合性制約を有効化するスクリプトが記述されています。

   このスクリプトには STATUS が DISABLE の整合性制約またはトリガーは含まれません。また INSERT 及び DELETE 操作に対する文／行トリガーが含まれます。

d. 環境変数 PARTITIONING（89 頁）へ文字列が設定されている時はパーティション名かサブパーティション名が制御ファイルのファイル名に追加されます。

e. データ・ファイルについても制御ファイル同様に環境変数 PARTITIONING（89 頁）へ設定された値の影響を受けます。

f. 拡張子に「blo」または「clo」がついているファイルはそれぞれ BLOB 型、CLOB 型（または NCLOB）の LOB 値を格納したファイルです。拡張子の名前は環境変数 EXTNAMECLOB（79 頁）及び EXTNAMEBLOB（79 頁）を設定することにより変更することが出来ます。ファイル名はアンダーバーで区切られた3つの部分文字列で構成されています。最初の文字列はその LOB 値が含まれていた表の名前を示します。2番目の文字列は「R」で始まり、以降10桁の数字になっており、LOB 値の含まれていた行を示します。3番目の文字列は「C」で始まり、以降4桁の数字になっており、LOB 値の含まれていた列を示します。

   clo ファイルは環境変数 CHARSETID（75 頁）に設定されたキャラクタ・セット ID によってエンコーディングが変化します。詳しくは「13.Unicode 対応について (125頁)」を参照してください。
   LOB 出力に関する詳細については「23.LOB 出力機能について（156 頁）」にある記事もご参照ください。

表の名前とは無関係でアンダーバーが前置されているファイル名を持つファイルは、SQL*Loader を介して表にデータを戻す際に利用できる、管理的作業の為のスクリプトが保存されています。以下は管理的作業の為のスクリプトに

ついての説明です。これらのファイルは環境変数 SKIPSCR（96 頁）の設定によって出力を制御することが出来ます。

g. _conn.sql にはロード予定先の接続情報が格納されています。このスクリプトは他の拡張子「sql」を持つスクリプトから SQL*Plus の「@」コマンドを使って呼び出されます。このファイルに含まれる CONNECT 文のパラメータは環境変数 DST_INFO（78 頁） へ指定した文字列が設定されます。

h. _disable_deps.sql は索引を削除し、整合性制約・トリガーを無効化するスクリプトです。STATUS が DISABLE の整合性制約またはトリガーは含まれません。INSERT 及び DELETE 操作に対する文／行トリガーが含まれます。環境変数 MNGSCROUT へ指定した値により文が実行されるかどうかを制御することが出来ます。_make.bat の中でデータロード後に実行されます。

i. _make.bat は出力したファイルを使って再ロード処理を一括実行するための Windows 用のシェル・スクリプトです。

j. _par0.txt は SQL*Loader でダイレクト・パス・ロードを実行させるためのコマンドライン・オプションが含まれたパラメータ・ファイルです。このファイルに含まれる USERID パラメータは環境変数 DST_INFO（78 頁） へ指定した文字列が設定されます。

k. _par1.txt は SQL*Loader で従来型パス・ロードを実行させるためのコマンドライン・オプションが含まれたパラメータ・ファイルです。このファイルに含まれる USERID パラメータは環境変数 DST_INFO へ指定した文字列が設定されます。

l. _wait_here.bat には再ロード時の機能的役割はありません。しかし_make.bat にその呼び出し文が記述されているのは、MPX によって_make.bat が実行された場合に、そのステップで並列実行中の全ての処理を同期化する目的があるためです。

この他、環境変数 LOADMETHOD（85 頁）に INSERT が指定されていた場合に限り、_trunc.sql というファイル名のスクリプトが出力されます。このファイルには出力対象表にまとめて truncate table を実行した後、可能であれば truncate table によって生じた追加エクステント解放後の断片化した空き領域のコアレス（結合）を行う処理が記述されています。

12.3. Query 機能

1. コマンドプロンプトを表示させ、製品フォルダ（43 頁）にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について（70 頁）」を参照して下さい。
2. Env.bat を保存した後、Query.bat を実行します。
3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>query
Multi-threaded Unloader version 4.2
                    :                                   :
                  「12.1.表示項目の共通部分(100頁」参照
                    :                                   :
任意問い合わせファイルを 1 個読込みました。(QUERYFIX) qryfix¥*.sql              (a)
任意問い合わせファイルを 2 個読込みました。(QUERYVAR) qryvar¥*.sql
バインド変数定義ファイルを読込みます。input¥bind.dat                            (b)
:b_job                          :SALESMAN
:b_sal                          :        1,300
:b_hiredate                     :1980-01-01
複数の接続を確立します。要求された並列度は 4 です。
データ・ファイルと制御ファイルを出力中です。output
TableName                       :      NumRows:  Elapse:Rows/s:Bytes: kB/s:   Len  (c)
```

```
SQL*Loader のコマンド行作成    :         3:    0.01
DEPT                          :         4:    0.02:    224: 128.:     7:    32.
EMP                           :         2:    0.02:    124: 151.:     9:    75.
SAL_AVG                       :         3:    0.01:    231: 113.:     8:    37.
処理対象となった表の総数      :         3                                         (d)
ラウンドトリップ（回）        :        76
出力バイト総数（kbytes）      :      1.96
所要時間（秒）                :      0.86
処理速度（kbytes/秒）         :      2.27
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. 任意問い合わせが含まれているファイルを環境変数 QUERYVAR(92 頁)、QUERYFIX(91 頁) の示すディレクトリから読み込みます。CSVと固定長データ形式のどちらで出力するかに応じて任意問合せファイルの置き場所を何れかに決めます。環境変数 QUERYFILENAME(93 頁)や、環境変数 EXTNAMESQL(80 頁)に任意文字列が指定されているとその文字列にマッチする名前のファイルだけを読み込みます。文字列にはワイルドカード文字(半角アスタリスクと半角クエスチョンマーク)を含めることができます。使用しない任意問い合わせファイルは拡張子を(例えば .bak などと)改名しておけば Query はこのファイルの読込をスキップします。任意問い合わせが含まれているファイルの名前は半角30文字以内で指定して下さい。出力結果の CSV ファイルを特定のテーブルに再ロードすることが予めわかっている場合には任意問い合わせファイルの名前にはそのテーブルの名前を付けておくとよいでしょう。またこのファイル名に半角アルファベット文字を使用する場合には大文字を使用したほうがよいでしょう。

b. 入力変数が定義されているファイルの場所を表示します。環境変数 INPUT(83 頁)の示すディレクトリ位置にある環境変数 FILEBIND(80 頁)の示す名前のファイル名から、バインド変数の定義情報を取得します。バインド変数は key=value の形式で定義します。key にはバインド変数名、value には変数の値を記述します。value を単一引用符(シングルクオーテーション)で囲むとその変数は文字列型として取り扱います。単一引用符で囲まれていなければその変数は整数型として取り扱います。固定小数点、浮動小数点の値をバインド変数へ与えることは出来ません。日付型の値は任意問い合わせ文に to_date 関数を記述して、文字列型値を日付型値へ変換して与えてください。環境変数 DATE_MASK(77 頁)と日付値の書式が一致してる場合は暗黙の型変換が行われるため to_date 関数は必要ありません。

c. データ・ファイルと制御ファイルの出力対象になっている表の以下の属性を示します。表 18 UnLoad 時に表示される情報(104 頁)を参照してください。

d. 各属性のバイト数には環境変数 STREAM_LOCATOR(98 頁)、ENCLOSER, ENCLOSER2(78 頁) に指定した区切文字(または区切文字列)及び改行文字の文字数も含みます。1秒あたりの平均出力行数の端数は小数点第1位以下が切り捨てられます。1行あたりの平均バイト数は小数点第1位以下が切り捨てられます。この平均バイト数はデータ・ファイルへ出力したときのサイズであって、Oracle 内部に格納されている状態の行のサイズと相関はありますが関数ではありません。従いましてこの値を領域サイズ見積もりに使用するのは適切でないので予めご了承の上お使いください。環境変数 BULK_SIZE(75 頁)と FEEDBACK(80 頁)を乗じた数の行数が取り出される度に表示が更新されます。FEEDBACK が 0 の場合、(c)は表示されません。また CTRL-C が入力され、プログラムの中断が要求されると FEEDBACK の値に関係なく「CAN」と表示されます。

e. データ・ファイルと制御ファイルの出力対象になった表の総数が表示されます。「正常終了しました。」が表示されたら環境変数 OUTPUT(89 頁)へ指定したディレクトリにデータ・ファイルやスクリプトファイルが出力されていることを確認して下さい。

12.4. クイック・エクスポート機能

12.4.1. MTU.xla アドイン

最初に使用するときだけ1から7までの手順を実施してください。

1. プログラムフォルダにある、MTU.xla を AddIns ディレクトリへコピーします。製品メニューの「コマンド・プロンプト」を表示して下記のコマンドを実行してください。

```
call platform.bat
copy ″%MTU_EXEC_DIR%MTU.xla″ ″%APPDATA%¥Microsoft¥addIns″
```

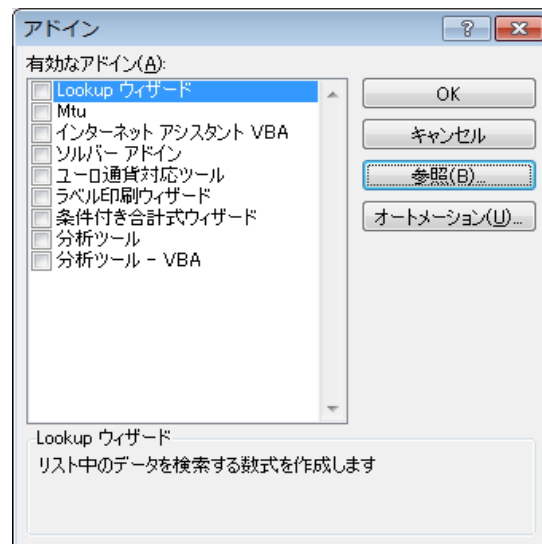
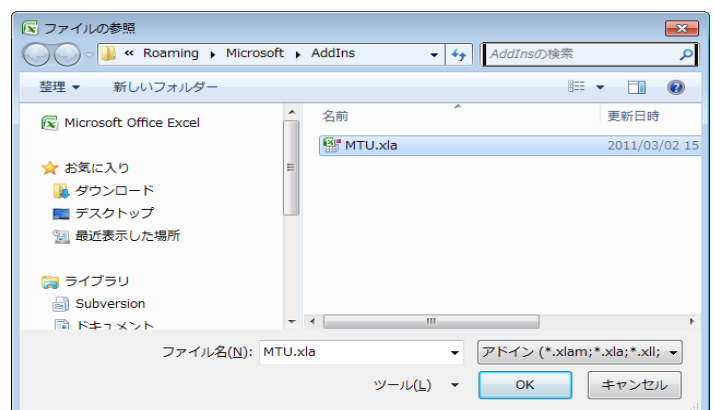
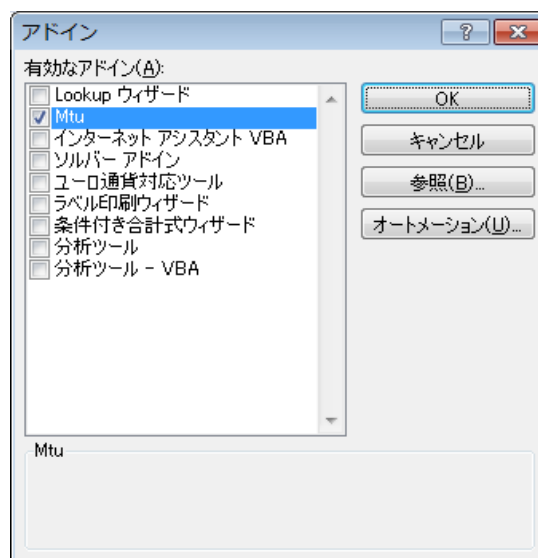
2. Microsoft Excel を起動します。
3. (Office Xp 以下の場合)メニューバーから[ツール]-[アドイン]を選択します。
4. (Office 2007 の場合)[Office ボタン]-[Excel のオプション]-[アドイン]-[Excel アドイン]-[設定]を選択します。
5. 表示されたアドインダイアログボックスの[参照]ボタンを選択します。

6. [ファイルの参照]コモンダイアログボックスから[MTU.xla]を選択します。

7. [アドイン]ダイアログボックスのリストに[MTU]というアイテムが追加され、チェックボックスがチェックされている事を確認し、[OK]ボタンを選択します。

8. Excel ユーバーへ新しく[マルチ・スレッド・アンローダー]というアイテムが追加された事を確認します。

図 2:Excel アドインメニュー (Office XP 以下)

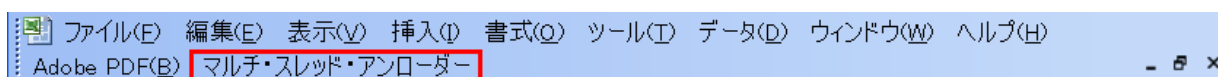



図 3::Excel アドインメニュー (Office 2007 以上)

9. メニューから[マルチ・スレッド・アンローダー]-[Quick Export]を選択します。[Oracle® SQL*Loader]というメニューは現在使用できません。

10. 続いて、図 4:MTU Quick Export ダイアログボックス(110 頁)が表示されます。

a. 接続先のユーザ名を入力してください。「25.プログラムの引数(25 頁)」の第一引数で指定する内容を入力してください。

b. 接続先のパスワードを入力してください。

c. 接続先の DB 別名を入力してください。

d. 問い合わせる表やビューが接続先とは異なるスキーマにある場合、その名前を入力してください。環境変数 SRC_USER(97 頁)へ設定する値と同一です。

e. 問合せの SELECT 文を入力してください。末尾には必ず半角セミコロンを入力してください。環境変数 QUERYVAR(92 頁)へ指定する値と同一です。

f. 問合せがプレースフォルダを含んでいる場合には、バインド変数の定義を key=value 形式で与えてください。プレースフォルダが複数種類あるときは定義をカンマで区切って複数与える事ができます。環境変数 FILEBIND(80 頁)へ指定する値と同一です。

g. 問い合わせたデータを格納する先のワークシートの名前を記述してください。

h. 実行形式ファイルが置かれている場所を入力するか、ボタンを押してコモンダイアログボックスで選択してください。

i. 実行時のログを格納する先のファイル名を入力してください。ボタンを押して格納先のディレクトリとファイル名を選ぶ事もできます。

j. [実行]ボタンは、全ての入力が終わった後、問合せを開始するために押す事ができます。

k. [ログ]ボタンは、実行時ログの内容を確認するためのダイアログボックスを開く事ができます。

l. [保存]ボタンは、このダイアログで入力した情報を次回も利用したいときに使います。

m. [閉じる]ボタンは、このダイアログを閉じる時に使います。

図 4:MTU Quick Export ダイアログボックス

[ページ2]について初期表示値を確認し、変更したいものがあれば入力してください。

n. Oracle DATE 型の書式を指定するのに使用します。環境変数 DATE_MASK(77 頁)へ指定する値と同一です。

o. Oracle TIMESTAMP 型、および TIMESTAMP WITH LOCAL TIME ZONE 型の書式を指定するのに使用します。環境変数 TIMESTAMP_MASK(98 頁)へ指定する値と同一です。

p. Oracle TIMESTAMP WITH TIME ZONE 型の書式を指定するのに使用します。環境変数 TIMESTAMP_TZ_MASK(98 頁)へ指定する値と同一です。

q. Oracle LONG 型を読取る時の最大バイト数を指定します。この値を超える部分のデータは切り捨てられます。環境変数 MAXLONGSIZE(85 頁)へ指定する値と同一です。

r. Oracle 浮動小数点型の書式を指定するのに使用します。環境変数 FPNUMFMT(83 頁)へ指定する値と同一です。

11. 入力値を確認後、[実行]ボタンを選択します。MTU が呼び出され、正常終了の場合、データがワークシートへ書き込まれます。データが大量に戻される場合は、処理が完了するまで画面がハングしたような状態になりますのでご了承ください。

	A	B	C	D	E	F	G	H
1	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
2	7499	ALLEN	SALESMAN	7698	1981/2/20 0:00	1600	300	30
3	7521	WARD	SALESMAN	7698	1981/2/22 0:00	1250	500	30
4	7654	MARTIN	SALESMAN	7698	1981/9/28 0:00	1250	1400	30
5	7844	TURNER	SALESMAN	7698	1981/9/8 0:00	1500	0	30

12. [ログ]ボタンを押すとMTUの画面出力をダイアログ表示する事ができます。下記にその事例を示します。

12.4.2. QikExp.bat スクリプト

1. コマンドプロンプトを表示させ、製品フォルダ(43 頁)にある、QikExp.bat スクリプトを実行します。

2. 以下に画面表示の説明をします。プロンプトに表示される角カッコ内の文字は入力を省略した時に、入力したものとみなされるデフォルト値です。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>QikExp
接続先 [SYSTEM/*]: SYSTEM/*@O111UTF8                                  (a)
スキーマ [SCOTT]:                                                     (b)
問合せ [select * into EMP from EMP]:                                  (c)
バインド変数 [:b_job="SALESMAN"]:                                     (d)
出力先 [output]:                                                      (e)
使用する区切文字 [0x09:0x22::]:                                       (f)
設定内容を保存しますか? ( y, n or q ) [n]: y                          (g)
設定内容を QikExp.dat へ保存しました。
Multi-threaded Unloader version 4.2
                             :                                :
                  「12.1.表示項目の共通部分(100頁」参照
                             :                                :
データ・ファイルと制御ファイルを出力中です。output
TableName                         :       NumRows:  Elapse:Rows/s:Bytes: kB/s:   Len
EMP                               :            14:    0.02:   691: 905.:    44:   64.
処理対象となった表の総数          :             1
ラウンドトリップ (回)             :            63
出力バイト総数 (kbytes)           :          1.51
所要時間 (秒)                     :         16.85
処理速度 (kbytes/秒)              :          0.09
```

a. 接続先の情報を入力します。「25.プログラムの引数(25 頁)」の第一引数で指定する内容を入力してください。

b. 前の手順で指定した接続先と異なるスキーマにある表をアクセスする場合にはそのスキーマ名を入力してください。環境変数 SRC_USER(97 頁)へ設定する値と同一です。

c. 問合せの SELECT 文を入力してください。環境変数 QUERYVAR(92 頁)へ指定する値と同一です。

d. 問合せがプレースフォルダを含んでいる場合には、バインド変数の定義を key=value 形式で与えてください。プレースフォルダが複数種類あるときは定義をカンマで区切って複数与える事ができます。環境変数 FILEBIND(80 頁)へ指定する値と同一です。

e. データ・ファイルの出力先フォルダを入力してください。環境変数 OUTPUT(89 頁)へ指定する値と同一です。

f. 列の値を区切るための文字セットを所定の書式で入力してください。環境変数 EMBED_COLUMN_NAME(78 頁)へ指定する値と同一です。

g. 3つの選択肢はそれぞれ下記の意味があります。状況に応じて適切なものを選択してください。

選択肢	意味
Y	入力した内容を次の実行の時に、デフォルト値として使えるよう QikExp.dat へ保存し、問合せを開始します。
N	入力した内容な保存せず、そのまま問合せを開始します。
Q	入力した内容な保存せず、問合せも中止します。

12.5. Reload 機能

1. コマンドプロンプトを表示させ、製品フォルダ(43 頁)にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について(70 頁)」を参照して下さい。

2. Env.bat を保存した後、Reload.bat を実行します。オプション引数を1つ指定することができます。この引数を指定したときの動作については「12.6.リロード時の接続先変更(116 頁)」にある記事をご覧下さい。

3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>reload
Multiplexed Process eXecutor version 1.06                              (a)
Copyright (c) 2007 - 2010 Plumsix Co.,Ltd. All rights reserved.
Parallel degree is 2.                                                  (b)
Processes will run with BELOW_NORMAL priority class.                   (c)
Show window mode is 6.                                                 (d)
"sqlldr,sqlplus" will be executed in parallel.                         (e)
Returned value from "findstr" will be ignored.                         (f)
Startup directory is "output".                                         (g)
STDOUT from the child processes will be stored to "_stdout.log".       (h)
Reading "_make.bat".                                                   (i)
Waiting.
    1/  10 completed.    3.33s elapsed.; sqlplus /nolog @_disable_deps  (j)
Synchronized.
    3/  10 completed.    3.79s elapsed.; sqlldr parfile=_par1.txt rows=10000
control=DEPT
    2/  10 completed.    3.80s elapsed.; sqlldr parfile=_par1.txt rows=10000
control=EMP
Waiting.
    4/  10 completed.    4.67s elapsed.; sqlldr parfile=_par1.txt rows=10000
control=SALGRADE
```

```
Synchronized.
    5/  10 completed.     1.20s elapsed.; sqlplus /nolog @EMP_DEPS
Waiting.
    6/  10 completed.     1.24s elapsed.; sqlplus /nolog @DEPT_DEPS
    7/  10 completed.     2.02s elapsed.; sqlplus /nolog @SALGRADE_DEPS
Synchronized.
    8/  10 failed with rc=1, but ignored.    0.40s elapsed.; findstr /n "ORA-" *.log
> _errm.txt
    9/  10 failed with rc=1, but ignored.    0.23s elapsed.; findstr /n
"SQL¥*Loader-" *.log >> _errm.txt
   10/  10 failed with rc=1, but ignored.    0.22s elapsed.; findstr /n "拒否されま
した。" *.log >> _errm.txt
The process spent 11.334s and passed over a prescribed section 1 time(s).        (k)
Completed successfully.                                                          (l)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. Reload.bat では製品にバンドルされている、弊社オリジナルユーティリティー「プロセス多重化実行ユーティリティー(通称:MPX、実行形式ファイル:mpx.exe)」が内部で呼び出されています。ここでは MPX のバナーが表示されているのでバージョンなどを確認してください。

表 19 MTU とバンドルされている MPX のバージョンの関係

MTU のバージョン	バンドルされた MPX のバージョン
V3.05/v3.06	v1.03
v4.2	v1.7

MPX の取り扱いについては製品ホームディレクトリにある「プロセス多重化実行ユーティリティー取扱説明書(ファイル名:readme_mpx.pdf)」をご覧下さい。

b. プログラム実行時の並列度が表示されます。この値は env.bat の環境変数 PARALLELISM(90 頁)へ設定された半分の値が採用されます。一般に Unload に比べ Reload はスケーラビリティーが低いので、Unload よりも小さくなるように設定しています。これを変更したい場合は Reload.bat 中の下記の計算式を変更してお使い下さい。

```
set /a MULTIPLE=%PARALLELISM%/2
```

c. MPX から起動されるプログラムの優先順位クラスが表示されます。この値は env.bat の環境変数 PRIORITY_CLASS(91 頁)へ設定されたものです。MPX で動作させる他のプログラムへの影響を考慮して決めてください。

他に優先すべきバッチ処理や、レスポンスタイムへの厳しい要求を満たす必要のある GUI プログラムが同じコンピュータ上で動作する場合、MORMAL 以上の優先順位クラスを与えるべきではありません。

d. MPX から起動されるプログラムの画面表示状態が表示されます。この値を変更するには Reload.bat のステートメント中の/sw スイッチに与える引数を書換えてください。

```
"%EXEC_MPX%" /pa %MULTIPLE% /pc %PRIORITY_CLASS% /sw 6 %SW_M% /i "findstr" /d "%OUTPUT%"
%SW_CSI% %SW_CSO% /c "_make.bat"
```

表 20 /sw スイッチの引数

引数		説明

0	SW_HIDE	プロセスのコンソールを表示しない
1	SW_NORMAL	プロセスのコンソールを表示する
6	SW_MINIMIZE	プロセスのコンソールを最小化する

e. MPX から起動されるプログラムの内、並列処理するものがカンマ区切りリストで表示されます。この値を変更するには Reload.bat のステートメント中の/m スイッチに与える引数を書換えてください。

```
set SW_M=/m "%EXEC_LOAD%,%EXEC_PLUS%"
```

Reload.bat で引用されている%EXEC_LOAD%、%EXEC_PLUS%は env.bat に定義がありそれぞれ sqlldr, sqlplus と読みかえられます。

f. MPX から起動されるプログラムの内、戻り値が非 0 を戻しても MPX を異常終了させないプログラムが表示されます。findstr という Windows の外部コマンドは検索対象が見つからないと非 0 を戻します。

従いまして Oracle エラーメッセージがログから検出された場合に MPX が異常終了しては都合が悪いので戻り値を無視するようにしています。

g. MPX から起動されるプログラムへ実行時に与えられるカレントディレクトリを示します。この値を変更するには Reload.bat のステートメント中の/d スイッチに与える引数を書換えてください。

```
"%EXEC_MPX%" /pa %MULTIPLE% /pc %PRIORITY_CLASS% /sw 6 %SW_M% /i "findstr" /d "%OUTPUT%"
%SW_CSI% %SW_CSO% /c "_make.bat"
```

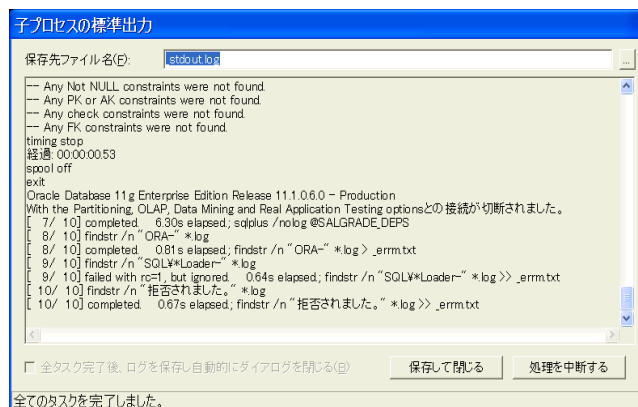
Reload.bat で引用されている%OUTPUT%は env.bat の環境変数 OUTPUT（89 頁）へ設定されたものです。

h. MPX から起動されるプログラムがコンソールアプリケーションである場合、その標準出力をテキストファイルへリダイレクトする事ができます。上記の例はそのリダイレクト先を表示しています。MPX はその他にも標準出力をダイアログボックスへ表示する事も可能です。その場合は Reload.bat 中の下記のステートメントを

```
rem set SW_CSO=
    set SW_CSO=/cso _stdout.log
```

コメントの場所を入替え、下記のように書換えてください。

```
    set SW_CSO=
rem set SW_CSO=/cso _stdout.log
```

上記設定により右記のようなダイアログボックスを表示する事も可能です。

i. MPX が実行するバッチスクリプトファイルが表示されます。この値を変更するには Reload.bat のステートメント中の/c スイッチに与える引数を書換えてください。/d が指定されていない場合、スクリプトファイルはカレントディレクトリにあるものと仮定されます。/d が指定されているとスクリプトファイルは/d に指定したディレクトリの下にあるものと仮定されます。

```
"%EXEC_MPX%" /pa %MULTIPLE% /pc %PRIORITY_CLASS% /sw 6 %SW_M% /i "findstr" /d "%OUTPUT%"
%SW_CSI% %SW_CSO% /c "_make.bat"
```

j. 前述のスクリプトファイルにあるステートメントの実行中／実行後の状態を示します。

- 左端の分数部分は分子に実行中ステートメントの行番号、分母にスクリプト全体の行数が示されます。
- 中ほどにある "N.NN elapsed."と表示された部分はステートメントの経過秒数が示されます。
- 右端の部分は実行されたステートメントが示されます。

k. MPX 全体の終了時点での経過時間他、統計情報が示されます。

l. MPX 終了時のステータスが表示されます。

表 21 MPX の終了時ステータス

スクリプト内のコマンド実行結果	表示されるステータス	MPX の戻り値
全て正常終了	Completed successfully.	0
ひとつ以上異常終了	Failed with an error.	非 0

12.6. リロード時の接続先変更

- Re.load.bat スクリプトにオプションの引数1つを指定することができます。これは、環境変数 DST_INFO（78 頁）で指定した当初のリロード先とは異なる接続先へデータをリロードしたい場合に指定します。
- 指定に当たっては次の書式で接続先文字列を与えてください。下の書式はローカル接続の場合です。

```
<username>/<password>@<db_alias>
または
<username>/<password>
```

- 接続先文字列に含まれる3つ（または2つ）の要素中で DST_INFO に指定したものと同じものは、省略することができます。変更したいものだけを記述してください。但し、リモート接続の場合の"/"と"@"、ローカル接続の場

合の”/”は省略することはできないのでご注意ください。

- Oracle のバージョンが 10g 未満の場合はこの機能を使用することができませんのでご注意ください。
- この機能は「パスワードの秘匿化について(128 頁)」に記載された、パスワード秘匿化機能とは組み合わせて使用することができませんのであらかじめご了承ください。
- リロード先を変更はあて先の指定を間違えると、回復困難なデータ消失を招くことがあります。繰り返しテストを行い、期待通りの動作であることを試験的な環境で十分確認してからご利用ください。

12.7. GetMeta 機能

1. コマンドプロンプトを表示させ、製品フォルダ(43 頁)にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について(70 頁)」を参照して下さい。
2. Env.bat を保存した後、GetMeta.bat を実行します。
3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>getmeta
Multi-threaded Unloader version 4.2
                                  :                                                          :
                  「12.1.表示項目の共通部分(100頁」参照
                                  :                                                          :
スキーマオブジェクト定義SQL文を出力中です。output¥meta.sql                              (a)
依存関係情報-Z                    :           6:    0.31                                    (b)
依存関係情報-A                    :          25:    0.47
セグメント属性                    :         172:    3.06
整合性制約情報-Z                  :           4:    0.03
整合性制約情報-A                  :          22:    0.43
一意キー制約の構成列名            :          24:    0.05
参照整合性制約の構成列名          :           1:    0.00
列属性                            :         342:    0.03
サブパーティション化キー列名      :           7:    0.17
パーティション化キー列名          :          12:    0.18
表サブパーティション属性          :          48:    0.09
表パーティション属性              :          28:    0.15
パーティション表属性              :           7:    0.04
表属性                            :          29:    0.21
関数ベース索引式                  :           4:    1.18
索引構成列名                      :          38:    0.12
索引サブパーティション属性        :          16:    0.43
索引パーティション属性            :          20:    0.43
パーティション索引属性            :          10:    0.03
索引属性                          :          22:    0.24
ハッシュ関数の定義                :           1:    0.23
クラスタ属性                      :           2:    0.08
クラスタ定義出力                  :           2:    0.00
親子関係情報-1                    :           6:    0.00
親子関係情報-A                    :          28:    0.38
クラスタキー列割当                :           4:    0.08
効力のあるキー制約                :          10:    0.07
子表が存在する親表                :           1:    0.13
制約（PK,AK,FK）の構成列名        :          25:    0.12
参照先制約構成列名                :          24:    0.04
```

```
外部キー制約                    :              1:    0.01
候補キー制約                    :             10:    0.00
LOB属性                         :              5:    0.30
列の省略時初期値                :             10:    0.02
トリガー属性                    :              2:    0.10
表の出力順序                    :             28:    0.01
表の定義出力                    :             28:    0.01
参照整合性制約名                :              1:    0.00
参照整合性制約の定義出力        :              1:    0.00
ビュー属性                      :              1:    0.46
ビューの定義出力                :              1:    0.00
処理対象となった表の総数        :             28:    0.03                  (c)
ラウンドトリップ（回）          :            282
出力バイト総数（kbytes）        :          48.17
所要時間（秒）                  :          14.72
処理速度（kbytes/秒）           :           3.27
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. ファイルの出力先を表示します。環境変数 OUTPUT(89 頁)と FILEMETA(81 頁)へ設定された値が表示されます。次の環境変数は GetMeta の動作に影響を与えるものです。それぞれの詳しい解説は「11. 環境変数について(70 頁)」を参照して下さい。

表 22 GetMeta の動作に影響を与える環境変数

変数名	参照頁
CMNTLVL	76
COMPRESS	76
DEPENDENT	77
METADIV	85
NMENCLS	87
NUMDAYS_AUDIT	88
PRINTCOLID	91

b. ファイルへ出力する情報をデータディクショナリから読み出す過程で、読み込んだ情報の種類と行数を表示します。環境変数 BULK_SIZE と FEEDBACK を乗じた数の行数が取り出される度に表示が更新されます。FEEDBACK が 0 の場合、(b)は表示されません。

c. 定義 SQL 文の出力対象になった表の総数が表示されます。「正常終了しました。」が表示されたら環境変数 OUTPUT と FILEMETA へ指定したディレクトリに定義 SQL 文の含まれたスクリプトファイルが出力されていることを確認して下さい。

12.8. その他の機能

12.8.1. ConsDisable 機能

1. コマンドプロンプトを表示させ、製品フォルダ(43頁)にあるEnv.batをテキストエディタで編集します。Env.batに設定されている環境変数に関する説明は「11. 環境変数について(70頁)」を参照して下さい。

2. Env.batを保存した後、ConsDisable.batを実行します。

3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>ConsDisable
Multi-threaded Unloader version 4.2
                    :                                    :
                    「12.1.表示項目の共通部分(100頁」参照
                    :                                    :
       2 個のデータベース・トリガーを無効化しました。                          (a)
       0 個の検査制約を無効化しました。                                        (b)
       1 個の参照制約を無効化しました。                                        (c)
       9 個の主キー／一意性制約を無効化しました。                              (d)
ラウンドトリップ（回）          :              33
所要時間（秒）                  :            6.62
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

この機能は環境変数 SRC_USER で指定したスキーマ内で現在有効(STATUS=ENABLED)な制約とデータベース・トリガーを無効化します。

a. 処理の結果、無効化されたデータベース･トリガーの合計数が表示されます。
b. 処理の結果、無効化された検査制約の合計数が表示されます。ただし Not NULL 制約とビュー制約は含みません。
c. 処理の結果、無効化された参照制約の合計数が表示されます。
d. 処理の結果、無効化された主キー制約と一意キー制約の合計数が表示されます。

「正常終了しました。」が表示されたら、データディクショナリ表の「ALL_CONSTRAINTS/ALL_TRIGGERS」を使って制約の「STATUS」が変更されたことを確認することが出来ます。事例を下記に示します。

```
define OBJOWNER=環境変数 SRC_USER へ指定した値
select constraint_name,constraint_type,status from all_constraints where owner =
'&OBJOWNER';
select trigger_name,trigger_type,status from all_triggers where owner = '&OBJOWNER';
```

12.8.2. ConsEnable 機能

1. コマンドプロンプトを表示させ、製品フォルダ(43頁)にあるEnv.batをテキストエディタで編集します。Env.batに設定されている環境変数に関する説明は「11. 環境変数について(70頁)」を参照して下さい。

2. Env.batを保存した後、ConsEnable.batを実行します。

3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>ConsEnable
Multi-threaded Unloader version 4.2
                    :                                    :
```

```
                    「12.1.表示項目の共通部分(100頁」参照
                    :                                          :
        9 個の主キー／一意性制約を有効化しました。                                        (a)
        1 個の参照制約を有効化しました。                                                  (b)
        0 個の検査制約を有効化しました。                                                  (c)
        2 個のデータベース・トリガーを有効化しました。                                    (d)
ラウンドトリップ (回)            :            41
所要時間 (秒)                    :          7.87
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

この機能は環境変数 SRC_USER で指定したスキーマ内で現在無効(STATUS=DISABLED)な制約とデータベース・トリガーを有効化します。

a. 処理の結果、有効化された主キー制約と一意キー制約の合計数が表示されます。
b. 処理の結果、有効化された参照制約の合計数が表示されます。
c. 処理の結果、有効化された検査制約の合計数が表示されます。ただし Not NULL 制約とビュー制約は含みません。
d. 処理の結果、無効化されたデータベース・トリガーの合計数が表示されます。

「正常終了しました。」が表示されたら、データディクショナリ表の「ALL_CONSTRAINTS/ALL_TRIGGERS」を使って制約の「STATUS」が変更されたことを確認することが出来ます。事例を下記に示します。

```
define OBJOWNER=環境変数 SRC_USER へ指定した値
select constraint_name,constraint_type,status from all_constraints where owner =
'&OBJOWNER';
select trigger_name,trigger_type,status from all_triggers where owner = '&OBJOWNER';
```

12.8.3. GetFkrb 機能

1. コマンドプロンプトを表示させ、製品フォルダ(43 頁)にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について(70 頁)」を参照して下さい。
2. Env.bat を保存した後、GetFkrb.bat を実行します。
3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>GetFkrb
Multi-threaded Unloader version 4.2
                    :                                          :
                    「12.1.表示項目の共通部分(100頁」参照
                    :                                          :
外部キー再構成スクリプトを出力中です。output¥fkrb.sql                                     (a)
整合性制約情報-Z                 :            22:     0.00                                (b)
整合性制約情報-B                 :             4:     0.11
外部キーを持つ表                 :             1:     0.18
列属性                           :             8:     0.16
```

```
制約（PK,AK,FK）の構成列名            :                1:    0.20
参照先制約構成列名                    :                5:    0.18
効力のあるキー制約                    :                1:    0.25
外部キー制約                          :                1:    0.03
ラウンドトリップ（回）                :               49
出力バイト総数（kbytes）              :             0.28
所要時間（秒）                        :             2.69
処理速度（kbytes/秒）                 :             0.10
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. ファイルの出力先を表示します。環境変数 OUTPUT と FILEFKRB（81 頁）へ設定された値が表示されます。
b. ファイルへ出力する情報をデータディクショナリから読み出す過程で、読み込んだ情報の種類と行数を表示します。環境変数 BULK_SIZE と FEEDBACK を乗じた数の行数が取り出される度に表示が更新されます。FEEDBACK が 0 の場合、(b)は表示されません。

「正常終了しました。」が表示されたら環境変数 OUTPUT と FILEFKRB へ指定したディレクトリに外部キー再構成スクリプトの含まれたスクリプトファイルが出力されていることを確認して下さい。

12.8.4. GetTblP 機能

1. コマンドプロンプトを表示させ、製品フォルダ（43 頁）にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について（70 頁）」を参照して下さい。
2. Env.bat を保存した後、GetTblP.bat を実行します。
3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>GetTblp
Multi-threaded Unloader version 4.2
                  :                                  :
                  「12.1.表示項目の共通部分(100頁」参照
                  :                                  :
表間参照階層図（正展開）　を出力中です。output¥tblp.txt                   (a)
整合性制約情報-Z                      :                4:    0.04
整合性制約情報-B                      :                4:    0.24
親子関係情報-1                        :               28:    0.13         (b)
親子関係情報-A                        :               28:    0.54
再帰的検索の実行                      :                7                  (c)
処理対象となった表の総数              :                5                  (d)
ラウンドトリップ（回）                :               42
出力バイト総数（kbytes）              :             0.11
所要時間（秒）                        :             2.28
処理速度（kbytes/秒）                 :             0.05
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. ファイルの出力先を表示します。環境変数 OUTPUT と FILEPLSP（82 頁）へ設定された値が表示されます。
b. 参照される側とする側の相互関係を示した表（APLDDT02）へ行を挿入します。

c. 再帰的検索を行った回数を表示します。500 回検索が行われる度に表示が更新されます。FEEDBACK が 0 の場合、(c)は表示されません。

d. 表間参照階層図（正展開）の出力対象になった表の総数が表示されます。

e. 「正常終了しました。」が表示されたら環境変数 OUTPUT と FILETBLP へ指定したディレクトリに表間参照階層図（正展開）の含まれたファイルが出力されていることを確認して下さい。

出力されたファイル内の枝分かれ部分にある「?」記号は親表を参照する当該表の参照整合性制約が無効であることを示します。

12.8.5. GetTblN 機能

1. コマンドプロンプトを表示させ、製品フォルダ（43 頁）にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について（70 頁）」を参照して下さい。
2. Env.bat を保存した後、GetTblN.bat を実行します。
3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>GetTbln
                  :                                  :
                  「12.1.表示項目の共通部分(100頁」参照
                  :                                  :
表間参照階層図（逆展開） を出力中です。output¥tbln.txt                          (a)
整合性制約情報-Z              :            4:    0.00
整合性制約情報-B              :            4:    0.02
親子関係情報-1                :           28:    0.00                           (b)
親子関係情報-A                :           28:    0.07
再帰的検索の実行              :            7                                    (c)
処理対象となった表の総数      :            5                                    (d)
ラウンドトリップ（回）        :           42
出力バイト総数（kbytes）      :         0.11
所要時間（秒）                :         0.76
処理速度（kbytes/秒）         :         0.14
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. ファイルの出力先を表示します。環境変数 OUTPUT と FILETBLN（82 頁）へ設定された値が表示されます。

b. 参照される側とする側の相互関係を示した表（APLDDT02）へ行を挿入します。

c. 再帰的検索を行った回数を表示します。500 回検索が行われる度に表示が更新されます。FEEDBACK が 0 の場合、(c)は表示されません。

d. 表間参照階層図（逆展開）の出力対象になった表の総数が表示されます。

「正常終了しました。」が表示されたら環境変数 OUTPUT と FILETBLN へ指定したディレクトリに表間参照階層図（逆展開）の含まれたファイルが出力されていることを確認して下さい。

出力されたファイル内の枝分かれ部分にある「?」記号は当該表を参照する子表の参照整合性制約が無効であることを示します。

12.8.6. GetPlsP 機能

1. コマンドプロンプトを表示させ、製品フォルダ(43 頁)にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について(70 頁)」を参照して下さい。
2. Env.bat を保存した後、GetPlsP.bat を実行します。
3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>GetPlsP
Multi-threaded Unloader version 4.2
                    :                                    :
                   「12.1.表示項目の共通部分(100頁」参照
                    :                                    :
名前付きPL/SQL間参照階層図（正展開） を出力中です。output¥plsp.txt              (a)
依存関係情報-Z                :          25:    0.32
依存関係情報-A                :          25:    0.91
親子関係情報-C                :          62:    0.17                             (b)
再帰的検索の実行              :          65                                      (c)
処理対象となった表の総数      :          54                                      (d)
ラウンドトリップ（回）        :         100
出力バイト総数（kbytes）      :        1.43
所要時間（秒）                :        4.36
処理速度（kbytes/秒）         :        0.33
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. ファイルの出力先を表示します。環境変数 OUTPUT と FILEPLSP(82 頁)へ設定された値が表示されます。
b. 参照される側とする側の相互関係を示した表(APLDDT02)へ行を挿入します。
c. 再帰的検索を行った回数を表示します。500 回検索が行われる度に表示が更新されます。FEEDBACK が 0 の場合、(c)は表示されません。
d. 名前付き PL/SQL 間参照階層図(正展開)の出力対象になった名前付き PL/SQL の総数が表示されます。

「正常終了しました。」が表示されたら環境変数 OUTPUT と FILEPLSP へ指定したディレクトリに名前付き PL/SQL 間参照階層図(正展開)の含まれたファイルが出力されていることを確認して下さい。

図内の枝分かれ部分にある「?」記号は当該オブジェクトが無効であることを示します。

12.8.7. GetPlsN 機能

1. コマンドプロンプトを表示させ、製品フォルダ(43 頁)にある Env.bat をテキストエディタで編集します。Env.bat に設定されている環境変数に関する説明は「11. 環境変数について(70 頁)」を参照して下さい。
2. Env.bat を保存した後、GetPlsN.bat を実行します。
3. 以下に画面表示の説明をします。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>GetPlsN
Multi-threaded Unloader version 4.2
                    :                                    :
                   「12.1.表示項目の共通部分(100頁」参照
                    :                                    :
```

```
名前付きPL/SQL間参照階層図（逆展開） を出力中です。output¥plsn.txt                (a)
依存関係情報-Z                  :          25:    0.18
依存関係情報-A                  :          25:    0.49
親子関係情報-2                  :          62:    0.00                              (b)
親子関係情報-D                  :          62:    0.11
再帰的検索の実行                :          65                                       (c)
処理対象となった表の総数        :          54                                       (d)
ラウンドトリップ（回）          :         103
出力バイト総数（kbytes）        :        1.35
所要時間（秒）                  :        1.21
処理速度（kbytes/秒）           :        1.12
正常終了しました。

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

a. ファイルの出力先を表示します。環境変数 OUTPUT と FILEPLSN（81 頁）へ設定された値が表示されます。

b. 参照される側とする側の相互関係を示した表（APLDDT02）へ行を挿入します。

c. 再帰的検索を行った回数を表示します。500 回検索が行われる度に表示が更新されます。FEEDBACK が 0 の場合、(c)は表示されません。

d. 名前付き PL/SQL 間参照階層図（逆展開）の出力対象になった名前付き PL/SQL の総数が表示されます。

「正常終了しました。」が表示されたら環境変数 OUTPUT と FILEPLSN へ指定したディレクトリに名前付き PL/SQL 間参照階層図（逆展開）の含まれたファイルが出力されていることを確認して下さい。

図内の枝分かれ部分にある「?」記号は当該オブジェクトが無効であることを示します。

13. Unicode 対応について

MTU がサポートする Unicode 関連機能について説明します。

- UnLoad 機能と Query 機能を使用するときだけ、環境変数 CHARSETID（75 頁）に設定されたキャラクタ・セット ID が製品の動作に影響します。
- CLOB、CHAR、VARCHAR2、NCHAR、NVARCHAR2、LONG 型はデータベース・キャラクタ・セットとは独立して、プライマリ・データ・ファイル／LOB ファイル出力時のキャラクタ・セットを選択することが出来ます。
- 環境変数 CHARSETID に設定された値、あるいは環境変数 NLS_LANG に設定された値に係らず、NCLOB 型の内容が出力された NCLOB ファイル(拡張子 clo を持つファイル)は常に UTF-16BE[19]でエンコーディングされています。BOM はありません。
- CLOB 型の内容が出力された CLOB ファイル(拡張子 clo を持つファイル)は,エンコーディングが環境変数 CHARSETID に設定された値、あるいは環境変数 NLS_LANG に設定された値に依存します。BOM はありません。
- CHAR、VARCHAR2、NCHAR、NVARCHAR2、LONG 型のエンコーディングは環境変数 CHARSETID に設定された値、あるいは環境変数 NLS_LANG に設定された値に依存します。AL32UTF8, UTF8 が指定されている場合、データ・ファイルは UTF-8 で出力されます。環境変数 NOBYTEORDERMARK（87 頁）に文字列が設定されていると BOM を出力しません。何も設定されていない場合は BOM を出力」します。
- プライマリ・データ・ファイル、セカンダリ・データ・ファイル LOB ファイル以外のファイルは常にクライアントの NLS_LANG に依存するキャラクタ・セット(日本語 Windows では Shift-JIS)でエンコーディングされています。
- 環境変数 MERGE_LOBS_INTO_SDF（85 頁）に値が設定されている場合に出力される、セカンダリ・データ・ファイル（SDF）は NLS_LANG、CHARSETID の設定に関係なく、UTF-16BE でエンコーディングされます。BOM 有、BOM 無は環境変数 NOBYTEORDERMARK（87 頁）によって選択できます。

表にまとめると下記のようになります。

表 23 データ型と Unicode 対応

	CHARSETID が 0 の時[20]	CHARSETID が 873 (AL32UTF8)の時
CHAR,VARCHAR2, NCHAR, NVARCHAR2, LONG	プライマリ・データ・ファイルへ NLS_LANG 依存で出力	プライマリ・データ・ファイルへ UTF-8 で出力.。BOM は NOBYTEORDERMARK（87 頁）の設定に依存
CLOB	CLOB ファイルへ NLS_LANG 依存で出力	CLOB ファイルへ UTF-8 (BOM 無し)で出力
NCLOB	CLOB ファイルへ UTF-16BE (BOM 無し)で出力	
制御ファイル、シェル・スクリプト、有効化無効化スクリプト	常に NLS_LANG 依存(Shift-JIS)で出力	

[19] UTF-16BE の"BE"とは Big Endian の意味です。

[20] 日本語 Windows 上で CHARSETID=0 を指定すると、832 (JA16SJIS)を指定したのと同じ結果になります。

14. x64 ベースの Windows 上での使用

MTU は x64 ベースの Windows 上での使用をサポートすしています。製品に含まれる x64 ベースの実行形式ファイルを使用する場合は、特に準備は不要でそのまま使用できます。

ただし何らかの事情があって x86 ベースの実行形式ファイルを使用する場合には WOW64[21]という 32-bit エミュレーター環境で動作させなければならない為、いくつかの前提条件があります。この章では MTU を WOW64 の上で動作させる方法について説明します。

Oracle Database 用の Oracle ホームディレクトリが無い（＝Oracle データベースを動作させていない）x64 ベース Windows コンピュータをご使用になる場合、Oracle Database Client 10g for Microsoft Windows (32-bit)のみをインストールするだけで MTU が動作するようになりますので後続の作業はありません。

x64 ベース Windows 上に Oracle Database for Microsoft Windows (x64)用の Oracle ホームを構築し、同じマシンで 32-bit 版 MTU もお使いになる場合には、マルチ Oracle ホームを構成する必要があります。この場合には次のように作業を行ってください。

下記の順序で Oracle 製品をインストールします[22]。Oracle Database 10g を例にとって説明します。

表 24:マルチ・オラクル・ホーム構成時の順序

手順	内容
1	Oracle Database 10g for Microsft Windows (x64) ORACLE_HOME = C:¥Oracle¥product¥10.2.0¥db_1 Oracle ホーム名 = OraDb10g_home1
2	Oracle Database Client 10g for Microsoft Windows(32-bit) ORACLE_HOME =C:¥Oracle¥product¥10.2.0¥client_1 Oracle ホーム名 = OraClient10g_home1

環境変数 PATH を次のように設定します。手順 2 を実施することにより Oracle のセットアッププログラムにより PATH 変数へ C:¥Oracle¥product¥10.2.0¥client_1¥bin が挿入されますが、これを取り除きます。

```
PATH=C:¥Oracle¥product¥10.2.0¥db_1¥bin;%PATH%
```

次に、MTU の unload.bat や getmeta.bat スクリプトを下記のように編集します。黄色網掛け部分を追記します。

- 編集前

```
@echo off
rem ====================================================================
```

[21] WOW64 については Microsoft サポートオンラインにある文書番号 896456 を参照してください。URL: http://support.microsoft.com/kb/896456/ja

[22] 解説に使用している ORACLE_HOME と Oracle ホーム名は参考事例です。

```
rem Copyright (c) 2007 Plumsix Co.,Ltd. All rights reserved.
rem ====================================================================

call "%~dp0¥env"
"%EXECUTABLE%" "%userid%" UnLoad
IF ERRORLEVEL 1 pause
```

- 編集後

```
@echo off
rem ====================================================================
rem Copyright (c) 2007 Plumsix Co.,Ltd. All rights reserved.
rem ====================================================================

call "%~dp0¥env"
set ORG_PATH=%PATH%
set TNS_ADMIN=C:¥Oracle¥product¥10.2.0¥db_1¥network¥admin
PATH=C:¥Oracle¥product¥10.2.0¥client_1¥bin;%PATH%
"%EXECUTABLE%" "%userid%" UnLoad
IF ERRORLEVEL 1 pause
PATH=%ORG_PATH%
set TNS_ADMIN=
```

上記のように設定すると mtu.exe が OCI.DLL を PATH から探す時、C:¥oracle¥product¥10.2.0¥db_1¥bin よりも C:¥oracle¥product¥10.2.0 ¥client_1¥bin が先に見つかる為、MTU を使う時だけ x86 版の OCI.DLL を正しくロードできるようになります[23]。

また環境変数 TNS_ADMIN を明示することにより Oracle Database 用に既に構成済みの sqlnet.ora, tnsnames.ora を Oracle Client を Oracle Client でも共有することが出来ます。結果として Oracle Client を使用することによるマルチ ORACLE_HOME 化の影響を上記のスクリプトの範囲内に収めることが出来ます。

[23] Oracle for Windows の場合は%ORACLE_HOME%¥bin に oracle.key がありますので ORACLE_HOME を明示的に設定する必要なありません。

15. パスワードの秘匿化について

MTU はデータベース認証に対する外部パスワード・ストア（10g Release 1 以上）や OS 認証などの Oracle が提供する認証方法を使用して、パスワードを平文で与えることなく利用する事が可能です。それぞれの認証方法についての詳細は Oracle Database 製品マニュアル「セキュリティー・ガイド[24]」を参照してください。

パスワードの秘匿化をしない場合、MTU では下記の場所にパスワードの平文が置かれます。

表 25 平文パスワードの含まれるファイル

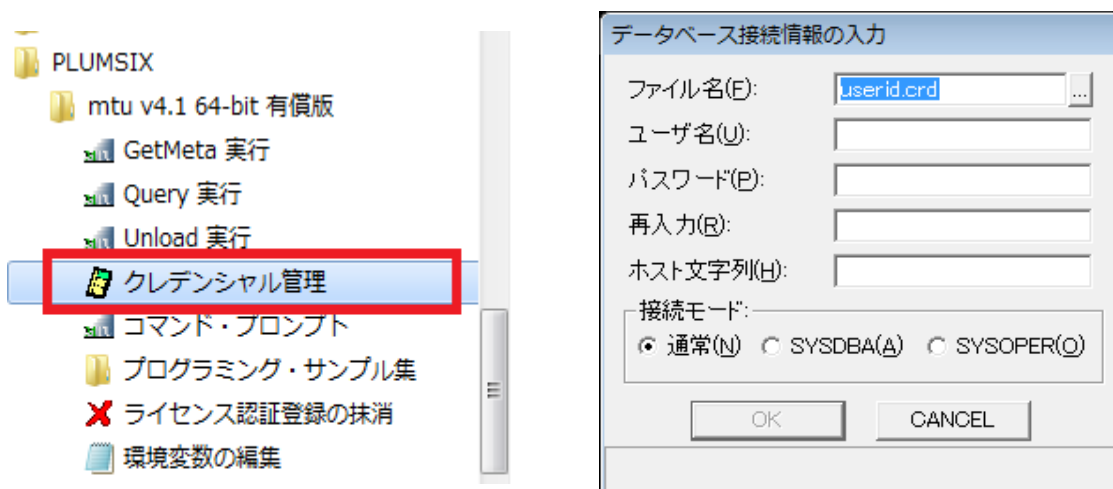
ファイルの場所と名前	説明
ENV.BAT	環境変数を定義する
output¥_conn.sql	再ロード時に SQL*Plus が読取る
output¥_par0.txt,output¥_par1.txt	再ロード時に SQL*Loader が読取る

システム構成上の都合で Oracle が用意する認証方法を利用する事ができないケースに対応するため弊社オリジナルのパスワード秘匿化機能を用意しています。これを利用すると、平文で書かれてしまう DB 接続の為のパスワードを、解読の困難な資格証明情報 (CRD) ファイルへ置き換えることが出来るようになります。

CRD ファイルは実行時に MTU、および MPX によって読取られます。MTU は読取ったデータの中から接続文字列を抜き出し復号化した後 DB 接続を試みます。

この機能を使うかどうかの設定は関係する環境変数の値の第一文字目にアットマーク”@”を使うか使わないかで切り替えられます。

15.1. CRD(クレデンシャル)ファイルの作成

パスワード秘匿化を行うには CRD ファイルを作成してください。CRD ファイル作成用のユーティリティーが用意されていますので、製品のメニューから「クレデンシャル管理」を選んでください。「データベース接続情報の入力」というダイ

アログボックスが表示されます。

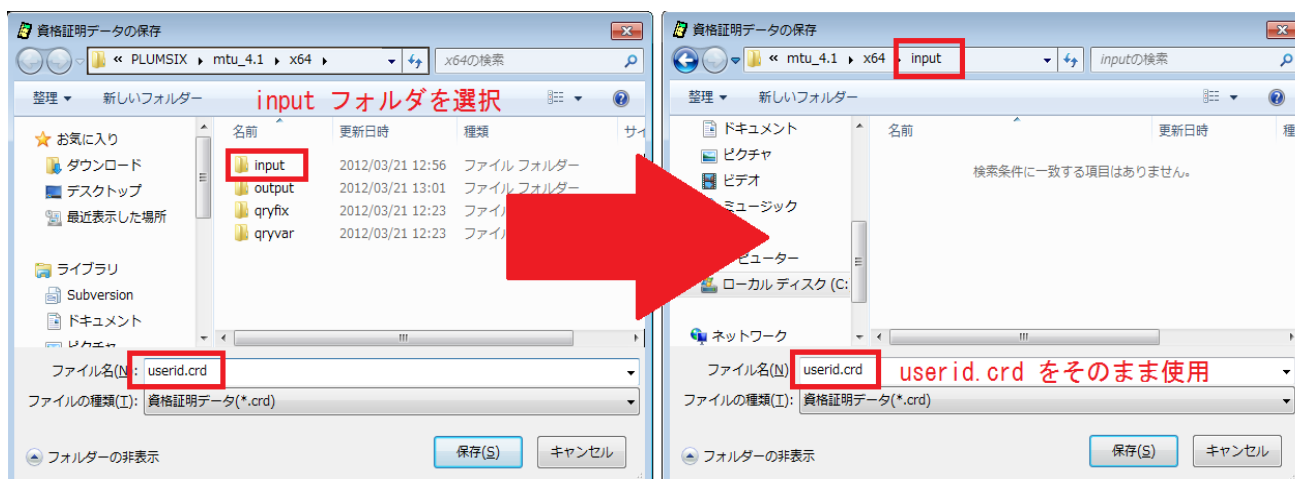
[24] Oracle Database セキュリティ・ガイド 11g リリース 2（11.2）「オペレーティング・システムを使用したユーザーの認証」、「パスワード資格証明用の安全性の高い外部パスワード・ストアの管理」

秘匿化対象として考えられるパスワードには下記の2つがあるので、必要に応じてそれぞれに対応する場所のディレクトリへ CRD ファイルを作成してください。どちらか1つだけを秘匿化して残りを従来通り平文で運用する方法も可能です。

表 26 秘匿化対象のパスワード

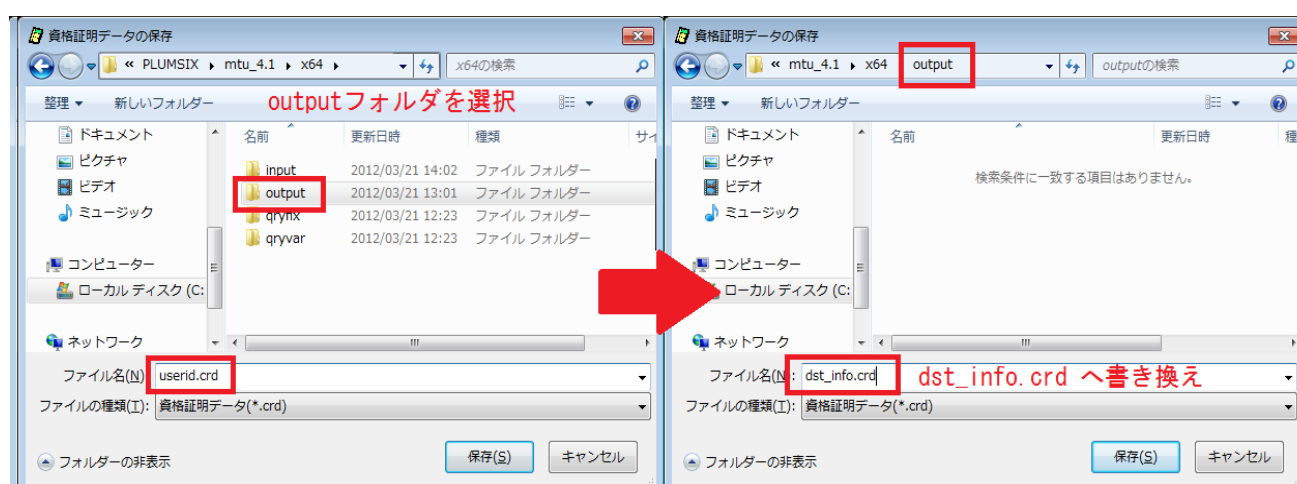
No	秘匿化対象	使われている場所	ディレクトリ	ファイル名の例
1	MTU が DB へ接続する時のパスワード	env.bat ファイル中の環境変数 USERID	環境変数 INPUT の指し示す先、例：input	userid.crd
2	MTU が出力したデータを再ロードする時に DB へ接続する時のパスワード	env.bat ファイル中の環境変数 DST_INFO	環境変数 OUTPUT の指し示す先、例：output	dst_info.crd

15.1.1. №1 の場合

「ファイル名」右側のボタンを押し、input フォルダを選択し、ファイル名を変えずにそのまま、保存ボタンを押してください。

この操作で、「ファイル名」へ選択したファイルがフルパスで表示されるようになります。

続いて「ユーザ」「パスワード」「ホスト文字列」へ必要事項を入力してください。接続先の Oracle Database のバージョンが 11g で、かつ初期化パラメータ SET SEC_CASE_SENSITIVE_LOGON =TRUE の場合はパスワードの大文字と小文字が区別されるので、字句通り入力してください。パスワード入力時は半角アスタリスク文字でエコーバックされます。接続モードは通常を選び、OK ボタンをおして入力内容を保存してください。

15.1.2. №2 の場合

「ファイル名」右側のボタンを押し、output フォルダを選択し、ファイル名を dst_info.crd へ書き換えて、保存ボタンを押してください。

この操作で、「ファイル名」へ選択したファイルがフルパスで表示されるようになります。

続いて「ユーザ」「パスワード」「ホスト文字列」へ必要事項を入力してください。接続先の Oracle Database のバージョンが 11g で、かつ初期化パラメータ SET SEC_CASE_SENSITIVE_LOGON =TRUE の場合はパスワードの大文字と小文字が区別されるので、字句通り入力してください。パスワード入力時は半角アスタリスク文字でエコーバックされます。接続モードは通常を選び、OK ボタンをおして入力内容を保存してください。

15.2. CRD ファイルの参照設定

前節の説明通りに mkcrd.exe ユーティリティーを使い CRD ファイルが出来ると、それを MTU や SQL*Plus、SQL*Loader が使うように設定する事ができます。エディタを使って環境変数設定の含まれている、env.bat ファイルを開き下記のように編集後保存してください。

表 27 P/W 秘匿化をする場合の env.bat 編集事例

環境変数	利用するプログラム	編集前	編集後
USERID	MTU	SYSTEM/MANAGER@o111utf8	@userid.crd
DST_INFO	SQL*Loader,SQL*Plus	SCOTT/TIGER@o111utf8	@dst_info.crd

従来ユーザ名／パスワード@DB別名を記述していた部分の代わりに、一文字目が半角アスタリスクで始まる、CRD ファイルのパスを指定します。アットマークがパスワード秘匿化機能を使用するという指定として解釈されます。

アットマーク直後のパスにディレクトリ名が含まれない場合のデフォルトのディレクトリは次の通りです。

- MTU の場合..環境変数 INPUT（83 頁）で指定した先
- SQL*Plus/SQL*Loader の場合..................環境変数 OUTPUT（89 頁）で指定した先

パスには、相対ディレクトリや絶対ディレクトリ、UNC などを含むことが可能です。

16. ダイレクト・パス・ロード

ダイレクト・パスロードは大量初期データの一括を短時間に終わらせたい場合に役立つ機能です。ダイレクトではない「従来」パスロードでは REDO ログが大量に発生する為、log file sync などの待機イベントで所要時間が長期化することがよくあります。

ただし、少量のデータを大量データを含むテーブルへロードする場合は索引をリビルドするコストが高くなりすぎるため、ダイレクト・パス・ロードの選択が適切ではないケースもあります。

この章ではダイレクト・パス・ロードを使用するにあたっての留意点について解説します。

16.1. ダイレクト・パス・ロードの効果と確認

```
set verify off
define priv_kb=0
column CkB format 999,999,999 heading 'Current kB' new_value PRIV_KB
column DkB format 999,999,999 heading 'Diff KB'
select ss.value/power(1024,1) as CkB
, ss.value/power(1024,1) - &PRIV_KB as DkB
from v$statname sn
, v$sysstat ss
where sn.name = 'redo size'
and ss.statistic# = sn.statistic#
;
```

例えば従来パスモードで ReLoad を実行すると下記のような結果が戻されます。

```
  Current kB      Diff KB
------------ ------------
   1,512,073      595,899
```

ダイレクト・パス・ロードで ReLoad を実行すると下記のような結果が戻されます。Redo エントリのの発生量が激減していることが分かります。

```
  Current kB      Diff KB
------------ ------------
   1,519,710        7,635
```

16.2. UNRECOVERABLE 句利用の留意点

MTU は環境変数 RECOVERABLE（93 頁）と USUALPATH（99 頁）に値を設定せずに UnLoad か Query を行うと、ロードされたデータを REDO ログ・ファイルへ記録しない方式で高速に再ロードするための制御ファイルを作成することが出来ます。再ロードの高速化と引替えに受け取るリスクが容認できるものであるなら、UNRECOVERABLE 句指定は時間短縮の効果的ソリューションです。UNRECOVERABLE 句はダイレクト・パス・ロードを選択するときだけ指定することができます。

UNRECOVERABLE 句を指定してロードを実行した表に対して、トランザクション処理を実行する場合は、トランザクションを開始する前にデータベース全体の物理バックアップを取得してください。UNRECOVERABLE 句を指定してロードを実行した直後はその表をリカバリする為に使用できる物理バックアップが１つも無い状態と同じだからです。

UNRECOVERABLE 句を指定してロードされたデータの REDO レコードは REDO ログ・ファイルに記録されません。かわりに Oracle は、回復操作を無効にするために必要な REDO ログ（無効 REDO ログ）を生成します。

この為、再ロードの前に取得したデータ・ファイルのバックアップを使用して媒体回復を行った場合、当該セグメントが回復できず無効状態になります。無効状態のセグメントにアクセスしようとすると下記の事例のような例外が戻されます。

```
SQL> select * from emp;
select * from emp
              *
エラー行: 1: エラーが発生しました。
ORA-01578: ファイル番号 3,ブロック番号 51 で Oracle データ・ブロックに障害が
発生しました。
ORA-01110: データ・ファイル 3 : D:¥ORACLE¥ORADATA¥ORCL¥USR1.DBF
ORA-26040: データ・ブロックが NOLOGGING オプションを使用してロードされました。

SQL> SELECT substrb(OWNER,1,14) as OWNER
2   , substrb(SEGMENT_NAME,1,30) as SEGMENT_NAME
3   , SEGMENT_TYPE
4   , EXTENT_ID
5   FROM DBA_EXTENTS
6   WHERE file_id = 3
7   AND 51 BETWEEN block_id AND block_id + blocks - 1
8   ;

OWNER          SEGMENT_NAME                   SEGMENT_TYPE        EXTENT_ID
-------------- ------------------------------ ------------------ ----------
SCOTT          EMP                            TABLE                       1

SQL>
```

このような状態から回復するには、GetMeta 機能（117 頁）を使って当該オブジェクトの create 文を取得し、表や索引を再作成するか、アンロード時のデータ・ファイルが残されている場合には再ロードし直すことで回復できます。

いずれにしても前回の再ロードから媒体障害発生までの間にコミットされた DML 操作の結果は回復することが出来ません。回復できましたら最初の DML を実行する前に、次の媒体障害に備えて速やかにバックアップを取得してく

ださい。

リカバリ不可能な操作が行われた事によってバックアップを必要としているデータ･ファイルの識別には Recovery Manager (RMAN) の report unrecoverable コマンドが役に立ちます。下記にその事例を示します。

```
C:¥temp>rman target /

Recovery Manager: Release 10.2.0.3.0 - Production on 土 12月 27 17:20:52 2008

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

ターゲット・データベース: O10G2 (データベースID=1959426794)に接続されました

RMAN> report unrecoverable;

リカバリ不能な操作のためバックアップの必要なファイルのレポート
File Type of Backup Required Name
---- ----------------------- ----------------------------------
9    full or incremental
C:¥ORACLE¥ORADATA¥O10G2¥DATAFILE¥O1_MF_USERS_427SQ7WB_.DBF
```

Repot unrecoverable コマンドにより9番目のデータ･ファイルにリカバリ不可能な操作が行われた(つまり nologging モードの索引作成やダイレクトロード)ということが報告されていますので、引き続き backup コマンドを実行することによりこのデータ･ファイルをバックアップすることにします。

```
RMAN> backup incremental level 0 datafile 9;

backupが開始されました(開始時間: 08-12-27)
チャネル: ORA_DISK_1が割り当てられました
チャネルORA_DISK_1: sid=141 devtype=DISK
チャネル: ORA_DISK_2が割り当てられました
チャネルORA_DISK_2: sid=158 devtype=DISK
チャネル: ORA_DISK_3が割り当てられました
チャネルORA_DISK_3: sid=138 devtype=DISK
チャネル: ORA_DISK_4が割り当てられました
チャネルORA_DISK_4: sid=136 devtype=DISK
チャネルORA_DISK_1: 圧縮型増分レベル0のデータファイル・バックアップ・セットを開始し
ています
チャネルORA_DISK_1: バックアップ・セットにデータ・ファイルを指定しています
入力データ・ファイルfno=00009 名前
=C:¥ORACLE¥ORADATA¥O10G2¥DATAFILE¥O1_MF_USERS_427SQ7WB_.DBF
チャネルORA_DISK_1: ピース1(08-12-27)を起動します
チャネルORA_DISK_1: ピース1(08-12-27)が完了しました
ピース・ハンドル
=C:¥ORACLE¥FLASH_RECOVERY_AREA¥O10G2¥BACKUPSET¥2008_12_27¥O1_MF_NNND0_TAG20081227T17
2237_4OCSG
GDG_.BKP タグ=TAG20081227T172237 コメント=NONE
チャネルORA_DISK_1: バックアップ・セットが完了しました。経過時間: 00:00:55
backupが完了しました(完了時間: 08-12-27)

Control File and SPFILE Autobackupが開始されました(開始時間: 08-12-27)
ピース・ハンドル
```

```
=C:¥ORACLE¥FLASH_RECOVERY_AREA¥O10G2¥AUTOBACKUP¥2008_12_27¥O1_MF_S_674587413_4OCSJ6P
7_.BKP コメント=NONE
Control File and SPFILE Autobackupが完了しました(完了時間: 08-12-27)
```

再度、report recoverableコマンドを実行してリカバリ不可能なデータ・ファイルが無いかどうか確認します。

```
RMAN> report unrecoverable;

リカバリ不能な操作のためバックアップの必要なファイルのレポート
File Type of Backup Required Name
---- ----------------------- -----------------------------------

RMAN> exit
```

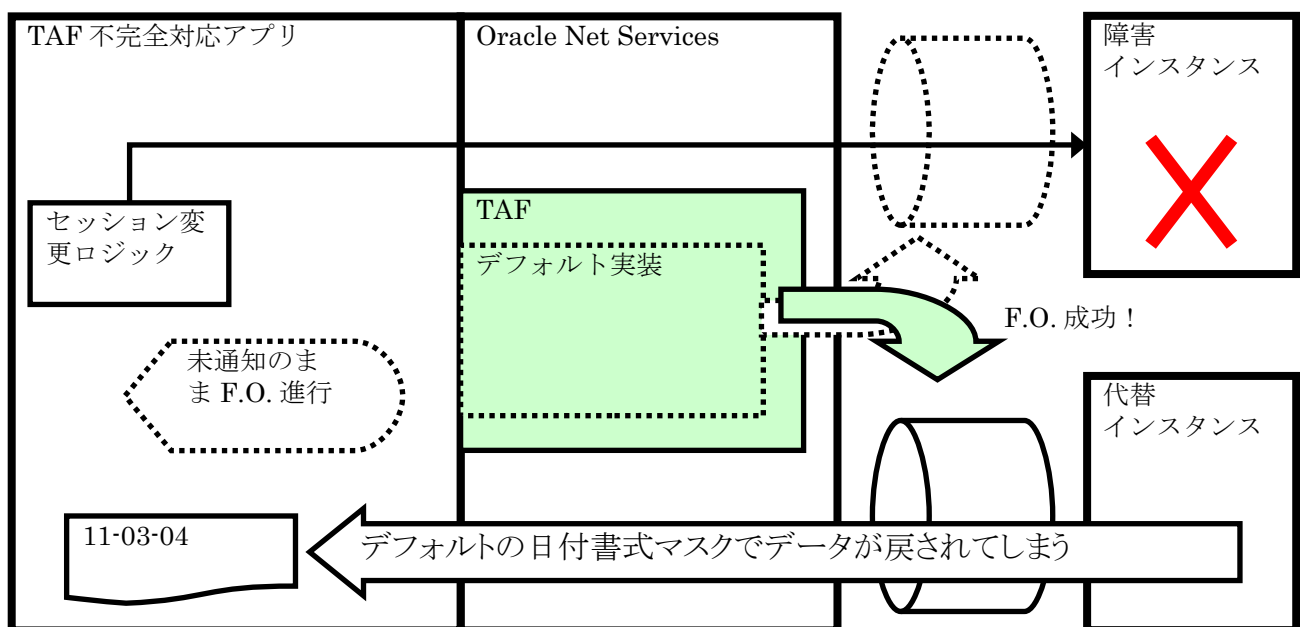
これで9番目のデータ・ファイルに含まれているデータはメディア障害から保護することが可能になりました。

17. TAF イベント対応

MTU は、TAF[25]と呼ばれている Oracle Database Client のオプションが適切に構成されている場合に自動的に適します。製品オリジナルの TAF イベントハンドラを OCI にフックするのでデフォルト実装のハンドらよりもより的確に、TAF イベントを処理できるようになっています。

デフォルト実装ハンドラをアプリケーションが使用する場合、図 5:TAF イベントに対応していないアプリケーション(136 頁)に示したように、運用上次の問題点が生じます。

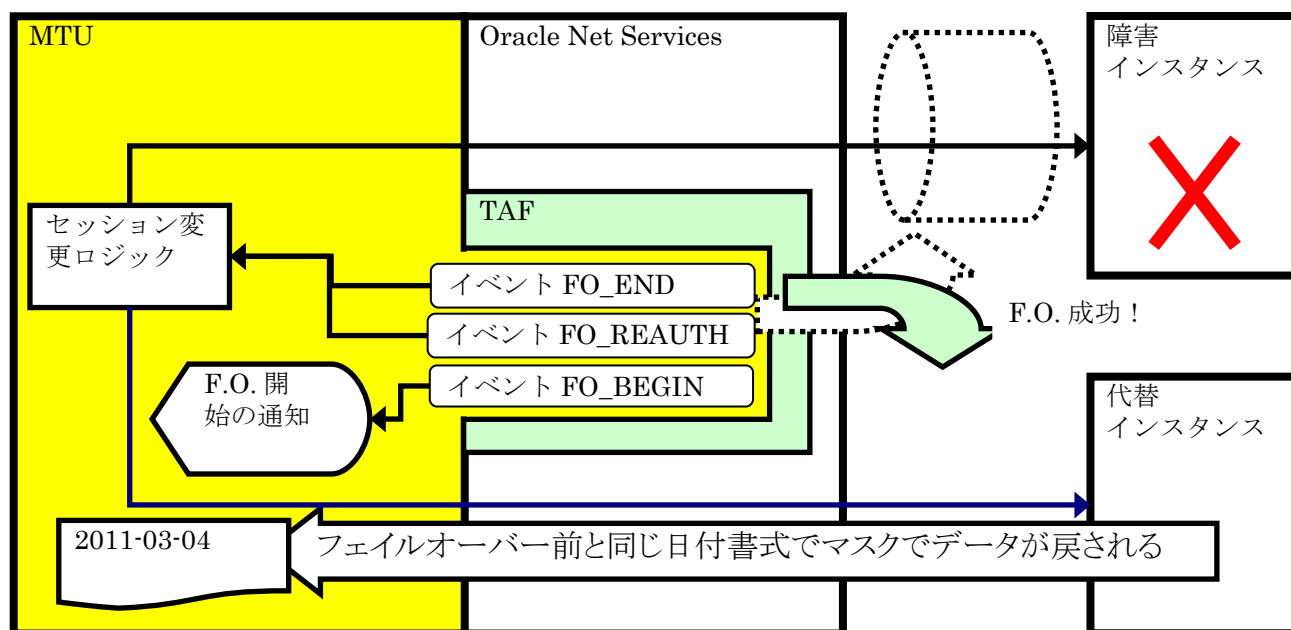
1. 接続先のインスタンス障害が発生しても、フェイル・オーバーが進行中かどうかがわからない。
2. フェイル・オーバーが完了した後、NLS_DATE_FORMAT 等のセッションパラメータがデフォルトに戻っている。

図 5:TAF イベントに対応していないアプリケーション

アプリケーションがこのような問題を抱えている場合、仮に TAF によって接続が維持され、カーソルの復元に成功したとしても現実には使い物になりません。

MTU はこれらの問題点を解決するため、製品固有の TAF イベントハンドラを備え、セッションが確立する際にこれらをサーバー・コンテキスト・ハンドルへフックします。セッションが終了するまでの間、TAF イベントが発生するとより現実的な運用の課題に対応できるよう、プロセスを制御します。具体的には次の点が可能となっています。

1. オペレーターへ接続先インスタンスの障害を素早く知らせること。
2. 進行中のフェイル・オーバーの進捗がわかること。
3. NLS_DATE_FORMAT 等のセッション・パラメータをフェイルオーバー後に自動的に復元すること。

[25] 透過的アプリケーション・フェイルオーバー

図 6:TAF イベントへ対応している MTU

接続先インスタンスの障害を検出すると、MTU は次のメッセージを表示して直ちにその事実をオペレータへ知らせます。このメッセージの出力の n 秒後に代替先インスタンスの探索が開始されます。何秒待機するかは環境変数 TAF_DELAY_TIME（98 頁）で指定することができます。

```
Information - [日付 時刻] Failing over (fo_type=SELECT) ... Please stand by.
```

RAC やアドバンスト・レプリケーションのように代替インスタンスがアクティブ運用の場合は n を比較的短時間に設定することができます。それとは反対に、HA や DataGuard のようにスタンバイ・インスタンスのアクティベーションに時間がかかるものは n を比較的長時間に指定してください。

代替先インスタンスが n 秒以上待機しても見つからない場合は次のメッセージが 5 秒おきに表示され、オペレータへ待機することを促します。

```
Information - [日付 時刻] Failover error gotten. Retried 回数 time(s). Sleeping.
```

代替先インスタンスが見つかると、次のメッセージが表示されセッションと中断していたカーソルのリカバリが開始されます。

```
Information - [日付 時刻] Failover ended ... resuming services.
```

カーソルがフェイル・オーバー前に大量の行を読み出していた場合、リカバリに時間がかかる場合があります。カーソルのリカバリに成功すると、中断していた箇所からの読み取りを再開し、フェイルオーバー前の動作に戻ります。

```
Information - [日付 時刻] Failover ended ... resuming services.
HASH_PART_TABLE              P2:      124,973:    8.31:    14k:   19M:2432k:   165.
HASH_PART_TABLE              P7:      124,904:    6.25:    19k:   19M:3232k:   165.
```

フェイルオーバーに失敗すると下記のメッセージが戻され、異常終了します。

```
Information - [日付 時刻] Failover aborted. Failover will not take place.
```

TAF 典型的な設定例を下記に示します。tnsnames.ora ファイルを次のように構成します。

```
RAC.PLUMSIX.LOCAL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = xxx.xx.xx.173)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = xxx.xx.xx.174)(PORT = 1521))
      (LOAD_BALANCE = yes)
    )
    (CONNECT_DATA =
      (SERVICE_NAME = RAC.plumsix.xxxxx)
      (FAILOVER_MODE =
        (TYPE = SELECT)
        (METHOD = BASIC)
      )
    )
  )
```

パラメータ FAILOVER_MODE はオプションですが TAF を構成する時には必須です。

MTU で TAF を使用したい場合はサブパラメータ TYPE は SELECT を指定してください。これにより、フェイルオーバーに成功すると読取が中断していた箇所から再開することができるようになります。サブパラメータ METHOD は BASIC か RRECONNECT のどちらでも利用できます。RETRIES, DELAY の両サブパラメータは設定があっても MTU は利用しません。

各パラメータの解説については Oracle 製品マニュアルの「Oracle Net Service 管理者ガイド」を参照してください。

18. 領域あふれ時のフェイルオーバー

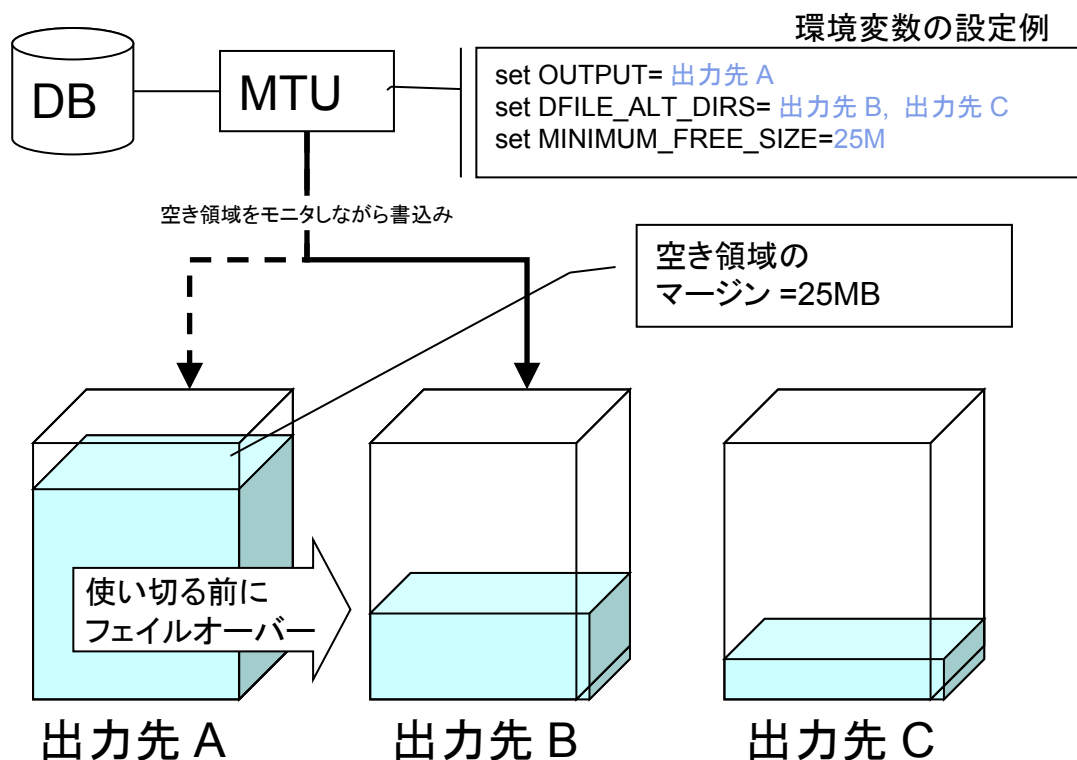
環境変数 MINIMUM_FREE_SIZE（86 頁）へ 1 以上（単位：バイト）の領域サイズを指定しておくと、出力先ディスクの空き領域不足の監視が可能になります。

実際の空き領域が指定された領域サイズを下回ると、環境変数 DFILE_ALT_DIRS（77 頁）にリストアップされたフォルダについて、リストの先頭から順番に空き領域をチェックします。

検討結果、十分な空き領域をもつ候補が見つかった場合は、それまでの出力先立ったファイルを閉じ、新たな出力先として選択されたフォルダへ新しいファイルが作成され、中断していた部分からデータ出力が再開されます。

すべての代替場所候補に空き領域が十分でない場合、自発的に異常終了します。

この機能は空き領域不足に敏感な他のアプリケーションと MTU を同居させる場合に役立ちます。

図 7:領域あふれ時のフェイルオーバー

空き領域をチェックする先には次のものがあります。

- 環境変数OUTPUT（89 頁）に指定した出力先
- 環境変数 AUXILIARY_DEST（74 頁）に指定した出力先
- 環境変数 DFILE_ALT_DIRS（77 頁）に指定した出力先

初期の書き込み対象はOUTPUT および AUXILIARY_DEST です。領域不足を検知すると DFILE_ALT_DIRS へカンマ

区切りリストとしてあらかじめ指定してある1つ以上の代替場所の左側にあるものから順にフェイルオーバーが試みられます。

フェイルオーバーに成功すると、フェイルオーバー前に書き込み対象であったファイルは即座にクローズされます。さらに第1回目のフェイルオーバーに限り拡張子を除くファイルの改名が行われます。改名後のファイル名は元のファイル名に半角アンダーバー記号と6桁の連番数値文字列を加えたものです。下記に例を示します。

```
改名前：NON_PART_LARGE_TABLE.dat
改名後：NON_PART_LARGE_TABLE_000001.dat
```

フェイルオーバーが繰り返された場合、6桁の連番数値の部分が1つずつインクリメントしたファイル名が与えられます。フェイルオーバーが完了すると、中断していた個所からデータ出力が再開されます。

フェイルオーバーの対象となるのはデータファイルに限ります。制御ファイルはフェイルオーバーされず元の場所への書き込みが続けられます。制御ファイルにはデータファイル名の参照が含まれているので、MTU は処理の最終段階で、フェイルオーバーにより出力先が変わったデータファイル名の参照場所の変更を行います。

DFILE_ALT_DIRS の各要素出力先はプログラム起動の初期段階で領域サイズがチェックされますので、実際に存在することと、空き領域が十分確保されていることを確認してください。

MINIMUM_FREE_SIZE に1以上が指定されて起動された場合、MTU は次の画面表示を行います。

```
空領域サイズチェックの閾値は xxx バイトです。
```

xxx の部分には MINIMUM_FREE_SIZE に指定した領域サイズが表示されます。また、DFILE_ALT_DIRS に要素出力先のカンマ区切りリスト（例えば"出力先 A,出力先 B,出力先 C"）が指定されている場合、MTU は次の画面表示を行います。

```
Information - 環境変数 DFILE_ALT_DIRS が指定されました。
  0: [287G] 出力先 A
  1: [372G] 出力先 B
  2: [98G] 出力先 C
```

出力先の前に表示された大括弧内の数値は現在のディスクの空き領域（単位：バイト）を示しています。Unload や Query の最中に領域不足が発生してフェイルオーバーに成功すると次のように表示されます。

```
Information - 出力先を切替えます。
  from: [12G] C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64¥output
    to: [287G] 出力先 A
```

"from:"がフェイルオーバー前に出力先だった場所、"to:"がフェイルオーバー後に出力先となった場所を示します。大括弧内の数値は切替え直後の空き領域サイズ（単位：バイト）を示しています。

さらに、制御ファイルの LOAD 句は次のように書きかえられます。

- フェイルオーバーが発生しない場合の制御ファイル

```
LOAD
INFILE "NON_PART_LARGE_TABLE" "VAR 10"
INTO TABLE "NON_PART_LARGE_TABLE"
```

- フェイルオーバーが発生した場合の制御ファイル

```
LOAD
INFILE "NON_PART_LARGE_TABLE_000001" "VAR 10"
INFILE "出力先A¥NON_PART_LARGE_TABLE_000002" "VAR 10"
INTO TABLE "NON_PART_LARGE_TABLE"
```

代替場所の候補が見つからない場合、次のメッセージを出力して MTU は異常終了します。xxxx には現在の出力先だったフォルダ名が表示されます。

```
Error - 空き領域サイズが十分ではありません。xxxxx
```

使用上の制限として DFILE_ALT_DIRS は環境変数 EMBED_COLUMN_NAME（78 頁）と併せて使用することができません。EMBED_COLUMN_NAME が設定されている場合は DFILE_ALT_DIRS は空とみなされます。

19. STREAM_LOCATOR の使用方法

環境変数は STREAM_LOCATOR は、環境変数 STDOUT（97 頁）と併せて使用します。STDOUT へ0以上の値を設定する事で標準出力へリダイレクトすると決めた出力対象を、ここで指定した別なストリームへリダイレクトすることが可能です。

STREAM_LOCATOR の値は次のような書式で指定して下さい。

```
スキーム名+”://”+出力先オブジェクト
```

スキーム名に指定可能な文字列は ipc_pipe と named_pipe です。

表 28: STREAM_LOCATOR に使用できるスキーム名

スキーム名	機能	参照先
ipc_pipe	並列化可能なパイプ処理	19.2.並列化可能なパイプ処理（144 頁）
named_pipe	名前付きパイプ	19.3.名前付きパイプ（146 頁）

19.1. 出力先オブジェクトの指定に使用可能なマクロ

出力先オブジェクトには OS システムのコマンドラインに使用することが可能な文字セットの他、向かいあう2つの半角中括弧文字'{'と'}'で囲まれた製品固有に定義されたマクロを指定することができます。マクロは MTU の実行時にマクロの種類と文脈に応じた文字列への置換えが行われます。マクロには次のものがあります。

表 29 :STREAM_LOCATOR 用マクロ

マクロ	意味
{O}	環境変数 OUTPUT（89 頁）に指定された文字列
{T}	Unload の時：表の名前 Query の時：クエリーの名前
{P}	Unload でパーティション表の時：パーティションの名前 Unload でヒープ構成表の並列アンロードの時：ROWID 範囲で分割された断片の2桁連番 Query で集合化クエリの時：各要素クエリの2桁連番
{C}	{T}+'_'+{P}
{A}	OUTPUT （89 頁）に指示した文字列か、AUXILIARY_DEST （74 頁）に指定した文字列。それぞれ領域あふれ時のフェイルオーバーが発生している場合、DFILE_ALT_DIRS（77 頁）に指定した文字列で現在選択されている代替先
{I}	環境変数 DST_INFO（頁）に指定された文字列
{X}	ファイル名拡張子

<table>
<tr><th>マクロ</th><th>意味</th></tr>
<tr><td></td><td>データファイルの場合:環境変数 DATAEXT へ指定した文字列、省略時は"dat"
制御ファイルの場合:"ctl"</td></tr>
<tr><td>{D=日付文字列}</td><td>Windows API GetDateFormat 関数の第 4 引数に指定できる文字列
<table>
<tr><th>制御文字</th><th>意味</th></tr>
<tr><td>d</td><td>年月日の日。1 桁の場合でも、先頭に 0 は付きません。</td></tr>
<tr><td>dd</td><td>年月日の日。1 桁の場合、先頭に 0 が付きます。</td></tr>
<tr><td>ddd</td><td>英語環境などでは、曜日の 3 文字の略称。日本語環境では (日)、(月) などになります。指定されたロケールに対応する LOCALE_SABBREVDAYNAME が使われます。</td></tr>
<tr><td>dddd</td><td>英語環境などでは、曜日の完全な名前。日本語環境では (日曜日)、(月曜日) などになります。指定されたロケールに対応する LOCALE_SDAYNAME が使われます。</td></tr>
<tr><td>M</td><td>月を表す数字。1 桁の場合でも、先頭に 0 は付きません。</td></tr>
<tr><td>MM</td><td>月を表す数字。1 桁の場合、先頭に 0 が付きます。</td></tr>
<tr><td>MMM</td><td>英語環境などでは、月名の 3 文字の略称。日本語環境では、1 月、2 月などになります。指定されたロケールに対応する LOCALE_SABBREVMONTHNAME が使われます。</td></tr>
<tr><td>MMMM</td><td>英語環境などでは、月名の完全な名前が返ります。日本語環境では、1 月、2 月などになります。指定されたロケールに対応する LOCALE_SMONTHNAME が使われます。</td></tr>
<tr><td>y</td><td>西暦の年の下 2 桁が返ります。1 桁の場合でも、先頭に 0 は付きません。</td></tr>
<tr><td>yy</td><td>西暦の年の下 2 桁が返ります。1 桁の場合、先頭に 0 が付きます。</td></tr>
<tr><td>yyy</td><td>西暦の年が 4 桁で返ります。</td></tr>
<tr><td>gg</td><td>王朝や元号を表す文字列。日本語環境では、「昭和」、「平成」などの元号が返ります。指定されたロケールに対応する CAL_SERASTRING が使われます。書式化対象の日付に王朝や元号などの情報が記録されていない場合、gg は無視されます。</td></tr>
</table></td></tr>
<tr><td>{W=時刻文字列}</td><td>Windows API GetTimeFormat 関数の第 4 引数に指定できる文字列
<table>
<tr><th>制御文字</th><th>意味</th></tr>
<tr><td>h</td><td>12 時間制の時間。1 桁の場合でも、先頭に 0 は付きません。</td></tr>
<tr><td>hh</td><td>12 時間制の時間。1 桁の場合、先頭に 0 が付きます。</td></tr>
<tr><td>H</td><td>24 時間制の時間。1 桁の場合でも、先頭に 0 は付きません。</td></tr>
<tr><td>HH</td><td>24 時間制の時間。1 桁の場合、先頭に 0 が付きます。</td></tr>
<tr><td>m</td><td>分。1 桁の場合でも、先頭に 0 は付きません。</td></tr>
<tr><td>mm</td><td>分。1 桁の場合、先頭に 0 が付きます。</td></tr>
<tr><td>s</td><td>秒。1 桁の場合でも、先頭に 0 は付きません。</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>マクロ</th><th>意味</th></tr>
<tr><td></td><td>
<table>
<tr><th>制御文字</th><th>意味</th></tr>
<tr><td>ss</td><td>秒。1 桁の場合、先頭に 0 が付きます。</td></tr>
<tr><td>t</td><td>1 文字の時刻マーカー。「A」「P」など。</td></tr>
<tr><td>tt</td><td>複数文字の時刻マーカー。「AM」「PM」「午前」「午後」など。</td></tr>
</table>
</td></tr>
<tr><td>{E=環境変数名}</td><td>環境変数に設定されている値</td></tr>
</table>

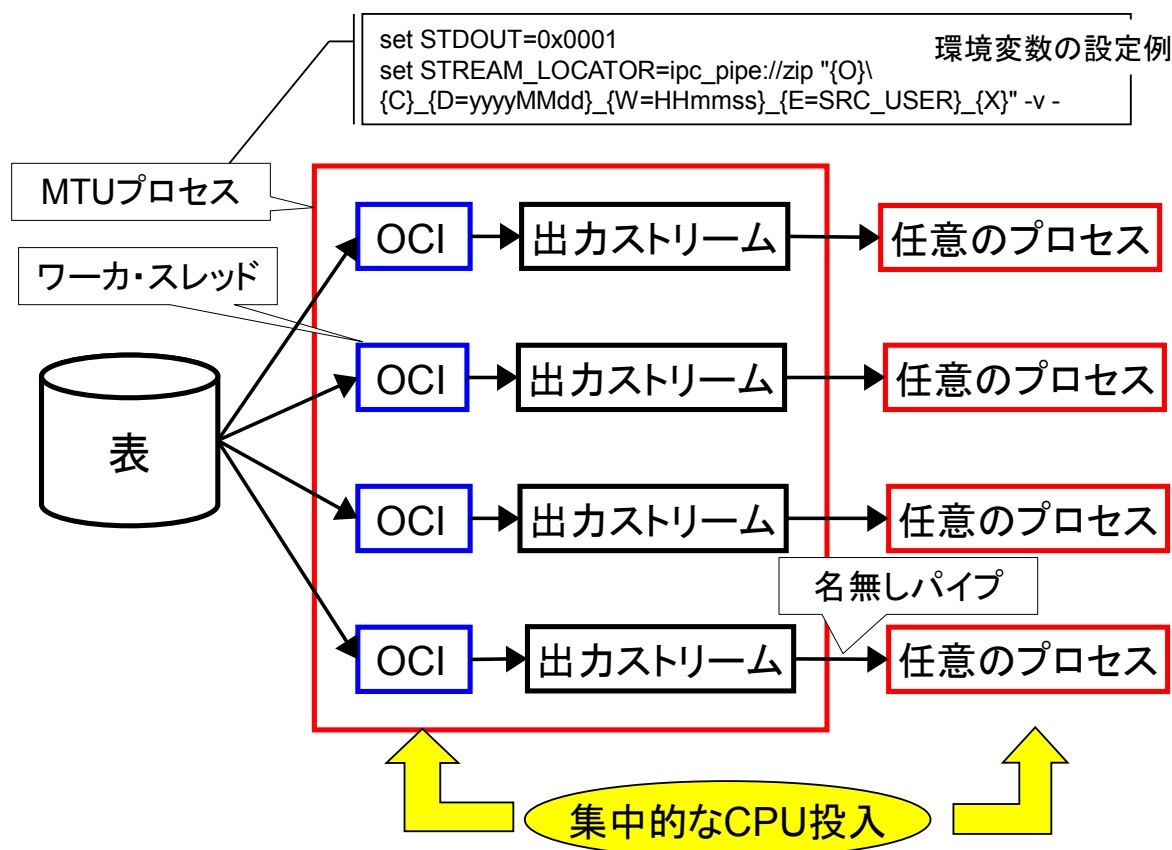
19.2. 並列化可能なパイプ処理

この機能はスキーム名に ipc_pipe を選択したときに使用できます。出力先オブジェクトに指定できるのは、そのコンピュータ上で実行できる、コンソールアプリケーションとその引数を併せた文字列です。一般的な Window アプリケーション[26]は、コンソールを明示的に割当てない限り[27]この機能と組み合わせることができません。

「並列化可能なパイプ処理」が適切に構成された場合、MTU とコンソール・アプリケーションが名無しパイプと呼ばれるプロセス間通信の手段を使ってダイレクトにデータを受け渡すようになります。下図参照。

[26] WinMain 関数がエントリポイントであるアプリケーション

[27] Window アプリケーションでも AllocConsole 関数を使ってコンソールの割り当てが可能です。

図 8:並列化可能なパイプ処理

データを中継するために一時なファイルを出力する必要が無いので領域使用とI/O 待ち時間削減という面で優れた効果を期待することが可能になります。

コンソールアプリケーション（＝任意プロセス）は MTU により、子プロセスとして起動されます。

起動後、MTU が用意した3本の名無しパイプと子プロセスが持つコンソール（標準入力、標準出力、標準エラー出力）がバインドされます。バインドが完了すると MTU は、次のような双方向通信を開始します。

- テキストデータ → 名無しパイプ → コンソールアプリの標準入力
- _stdout.log ← 名無しパイプ ← コンソールアプリの標準出力
- _stderr.log ← 名無しパイプ ← コンソールアプリの標準エラー出力

_stdout.log と_stderr.log は環境変数 OUTPUT（89 頁）へ指定した場所へ出力されます。これらのファイルは非同期的に更新されるので、処理継続中に書き込まれた内容はコンソールアプリの実際の処理フェーズよりも遅れているのでご注意ください。同期化されるのはコンソールアプリの終了したタイミングとなります。

また、環境変数 PARALLELISM（90 頁）が2以上である時、指定した数のコンソールアプリが同時に起動し_stdout.log と_stderr.log へそれぞれの出力を書き込みます。それぞれのコンソールアプリの出力を区別できるようにするため、MTU は次の書式の開始時と終了時のヘッダ情報を追加で書き込みます。

```
Information - [B:年-月-日 字:分:秒: マクロ展開後の出力先オブジェクト]
```

```
Information - [E:年-月-日 字:分:秒: マクロ展開後の出力先オブジェクト]
```

[B:]ヘッダと[E:]ヘッダの間にコンソールアプリがコンソールへ戻した表示が書き込まれます。出力先オブジェクトを見ることによりどのコンソールアプリの出力かを判断することができます。

MTU はパイプに送るべきデータが無くなると、名無しパイプをクローズします。コンソールアプリはこれを EOF(End of File)として受け取ることができます。

MTU はコンソールアプリが完全に終了するまで待機し、コンソールアプリの終了コードを受け取ります。受け取った終了コードが非 0 の場合は、MTU 自身のエラーとして扱い異常終了します。コンソールアプリの終了コードは MTU の終了コードに変換されます。この時、MTU の画面表示に

```
Error - プロセス ＜出力先オブジェクト＞ が異常終了しました。rc=x (cFileLocProc.c-yyy)
```

と表示された中の x がコンソールアプリの終了コードを示します。終了コードを外部から直接入手する方法はありません。

「並列化可能なパイプ処理」は後の項で説明する「19.3.名前付きパイプ(146 頁)」と用途が似通っていますが、違いを説明する為下表に長所をまとめます。短所は互いの長所の反対です。

表 30:各並列パイプ処理機能のまとめ

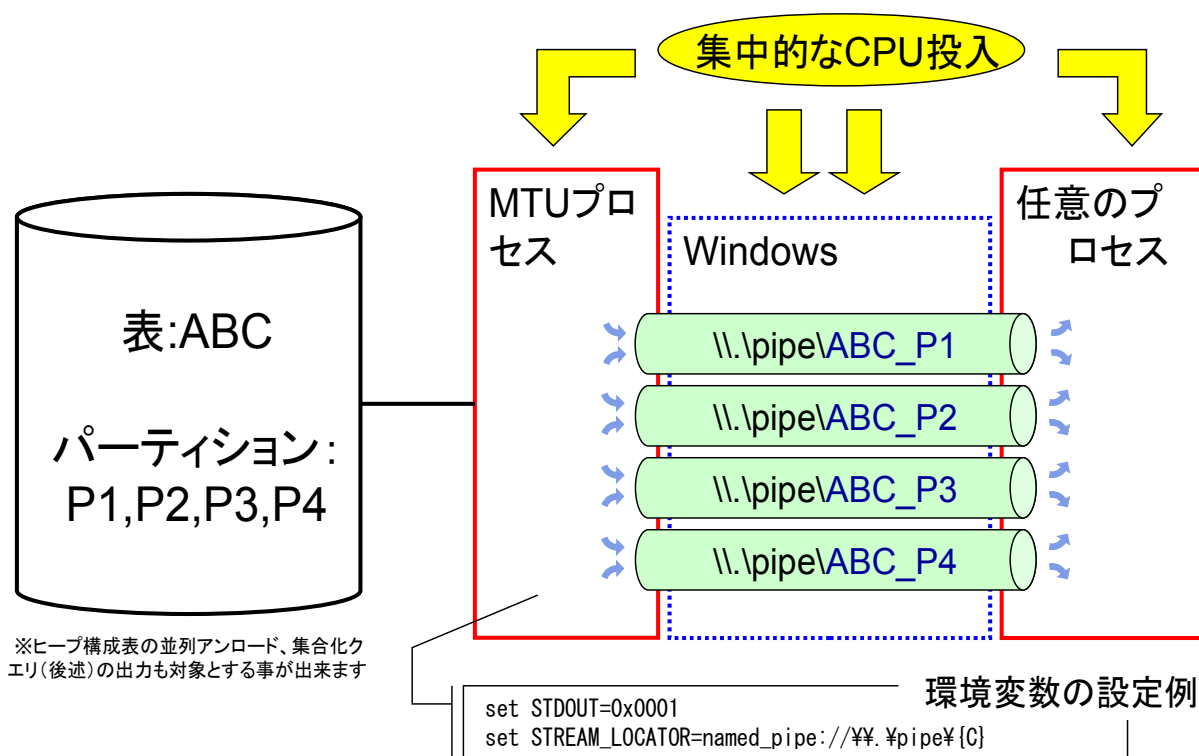
機能名称	長所
19.2.並列化可能なパイプ処理(144 頁)	● コンソールアプリを起動するタイミングは MTU が決定するので、コンソールアプリが先に起動したことによってオープンエラーが発生することは無い。 ● 並列プログラミング無しでマルチタスクが可能。 ● ”移動窓”(後述)を考慮したプログラミングが不要。
19.3.名前付きパイプ(146 頁)	● IPC の相手として指定できる任意プロセスは、コンソールアプリのみに限定されない。 ● 入力データのファイル名を指定できるアプリケーションであれば、替わりに名前付パイプの名前を与えることが可能。 ● 任意プロセスの戻り値を直接評価できるので、異常発生時の診断が容易。

19.3. 名前付きパイプ

この機能はスキーム名に named_pipe を選択したときに使用できます。出力先オブジェクトに指定できるのは、名前付パイプを表現する「¥¥.¥pipe¥名前」という書式の文字列です。名前の部分には、256 文字以内の円記号(¥)を除き、数字や記号を含む任意の文字を指定できます。名前付きパイプに関する説明は Microsoft 社の CreateNamedPipe 関数に関する説明を参照してください。

名前付きパイプはファイル名の替わりに使用できるので、コンソール・プログラムでなくても IPC の相手方としてつなげることができます。

「名前付パイプ」が適切に構成された場合、MTU と任意のプロセスが名前付きパイプと呼ばれるプロセス間通信の

図 9:名前付きパイプ

手段を使ってダイレクトにデータを受け渡すようになります。下図参照。

データを中継するために一時なファイルを出力する必要が無いので領域使用とI/O 待ち時間削減という面で優れた効果を期待することが可能になります。

MTU は自分自身の存続期間中「¥¥.¥pipe¥名前」の書式に基づく名前付パイプを提供します。

パイプの作成が終わると、同じ名前の名前付パイプをオープンしようつするプロセスがほかに現れるまで無限に待機します[28]。

MTU はパイプに送るべきデータが無くなると、名前付きパイプをクローズします。任意プロセスではこれを EOF として受け取ることができます。

MTU が異常終了した場合、例外処理で任意プロセスへ繋がっているパイプの強制切断を試みます[29]。任意プロセスはこの操作の結果 ERROR_PIPE_NOT_CONNECTED というエラー受け取るのでこれに対する例外処理を設けると MTU の異常終了に対処することができます。

反対に任意プロセスが異常終了した場合、MTU は「Code=0xe8, Msg=パイプを閉じています。」というエラーを受け取って直ちに異常終了します。

---

28 ConnectNamedPipe 関数を使用して待機します。

29 DisconnectNamedPipe 関数を使用して切断します。

MTU がパイプを作成する前に、同じ名前のパイプへ任意のプロセスが接続しようとするとパイプが見つからずにエラーとなります。従って、名前付パイプを使って MTU と IPC を行おうとする任意のプロセスは、MTU よりも後に起動させなければなりません。任意プロセスをパイプが見つかるまで待機させたければ、WaitNamedPipe 関数を使用して戻された ERROR_FILE_NOT_FOUND に関する例外処理を設けて下さい。

ファイル出力の場合は、待機せずにファイルを作成できますが、パイプの場合はオープンする相手が現れるまで待機するので、MTU のデータ処理の順序が問題となる場合があります。

環境変数 PARALLELISM（90 頁）へ2以上が設定され、MTU が並列処理可能な場合、名前付きパイプへのデータ書込みも並列に行うことが可能です。この時、出力対象の表やパーティションが複数ある場合、そのデータに関する名前付きパイプを作る順序は、セグメント・サイズの逆順です（下図参照）。

つまり、この逆順の名前リストが並列度数で切り取られる「移動窓」に対して任意プロセスがタイミングよく名前付きパイプをオープンしてこなかった場合、MTU の待機が発生します。

PARALLELISM=3 の場合　　移動窓：　　黒字：未出力／出力中　赤字：出力済み

開始

セグメント名	MB
TABLE_F	283
TABLE_E	221
TABLE_B	175
TABLE_A	63
TABLE_C	54
TABLE_D	23
TABLE_G	10

TABLE_B 完了

セグメント名	MB
TABLE_F	283
TABLE_E	221
TABLE_B	175
TABLE_A	63
TABLE_C	54
TABLE_D	23
TABLE_G	10

TABLE_A 完了

セグメント名	MB
TABLE_F	283
TABLE_E	221
TABLE_B	175
TABLE_A	63
TABLE_C	54
TABLE_D	23
TABLE_G	10

TABLE_E 完了

セグメント名	MB
TABLE_F	283
TABLE_E	221
TABLE_B	175
TABLE_A	63
TABLE_C	54
TABLE_D	23
TABLE_G	10

TABLE_F 完了

セグメント名	MB
TABLE_F	283
TABLE_E	221
TABLE_B	175
TABLE_A	63
TABLE_C	54
TABLE_D	23
TABLE_G	10

TABLE_G 完了

セグメント名	MB
TABLE_F	283
TABLE_E	221
TABLE_B	175
TABLE_A	63
TABLE_C	54
TABLE_D	23
TABLE_G	10

TABLE_D 完了

セグメント名	MB
TABLE_F	283
TABLE_E	221
TABLE_B	175
TABLE_A	63
TABLE_C	54
TABLE_D	23
TABLE_G	10

終了

セグメント名	MB
TABLE_F	283
TABLE_E	221
TABLE_B	175
TABLE_A	63
TABLE_C	54
TABLE_D	23
TABLE_G	10

この待機を発生させないようにする解決策には次の方法があります。

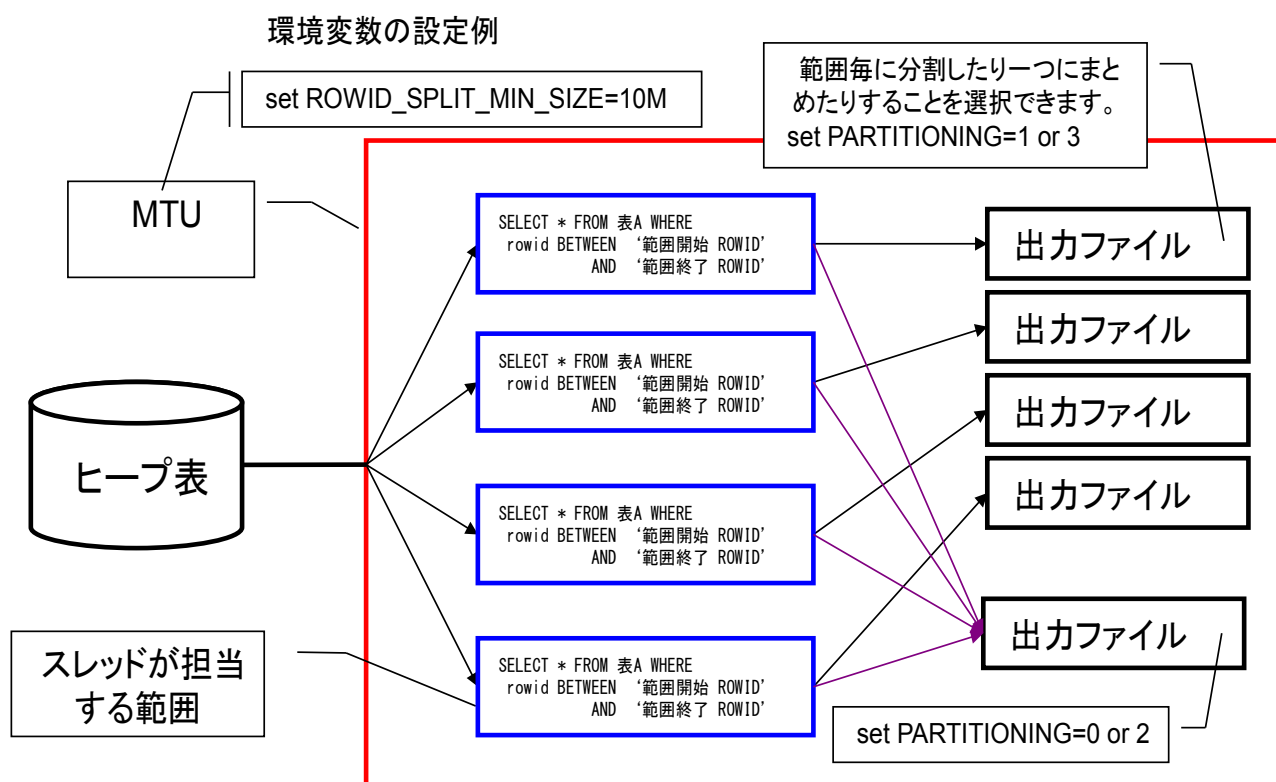
- 並列度をすべての出力対象セグメントの数以上に設定する（移動窓を広げる）。
- STREAM_LOCATION を設定せずに一度リハーサルを行い逆順リストを入手し、その順序通りに名前付きパイプを順次オープンするよう、任意プロセス側をプログラミングする。

- 任意プロセスで WaitNamedPipe 関数をコールするようプログラミングし移動窓がどこに開いていてもオープンエラーにならないようにする。

「名前付パイプ」は前の項で説明する「19.2.並列化可能なパイプ処理（144 頁）」と用途が似通っていますが、違いを説明する為「表 30:各並列パイプ処理機能のまとめ（146 頁）」に長所をまとめます。短所は互いの長所の反対です。

20. ヒープ構成表の並列アンロード

環境変数 ROWID_SPLIT_MIN_SIZE(94 頁)へ 1 以上(単位:バイト)の値を設定すると、ヒープ構成表を複数の ROWID ベースの範囲に分割してアンロードすることが可能になります。環境変数 PARALLELISM(90 頁)に 2 以上の値

図 10:ヒープ構成表の並列アンロード

が指定されている場合 16 を最大値として最適な並列度を選ぶことができます。

パーティション表、クラスタ表、索引構成表はこの機能の対象外です。

また、内部的にディクショナリビューの DBA_EXTENTS, DBA_OBJECTS を参照しますので、所有ユーザに対して、DBA ロールが割り当てられていない場合は次のコマンドを使って所有ユーザに対して権限を割り当てて下さい。コマンド等の詳しい内容については「8.5.システム権限の付与(45 頁)」にある記事を参照して下さい。

PARALLELISM の設定値が 16 を超えている場合は次のメッセージが表示され、分割数が 16 に制限されます。

```
Information - ROWID 分割の上限数は 16 です。
```

複数の ROWID 範囲に分割された行は範囲毎にデータファイルが作成されます。このとき、各データファイルに含まれる行数が必ずしも等分割にはならない事をご了承下さい。ヒープ構成表のエクステントが複数のデータファイルに分散して格納されている場合、あるいはブロックに含まれる行の分布やセグメントに含まれるエクステントのサイズが均

一でない場合このような現象が発生します。

環境変数 SORTING(96 頁)に任意文字列が指定されている場合は分割されません。次のようなメッセージが表示され、ROWID_SPLIT_MIN_SIZE には 0 が指定されたものとみなします。

```
Warning - 環境変数 SORTING に値を指定している場合は、環境変数 ROWID_SPLIT_MIN_SIZE
へ指定した値は無視され 0 バイトを指定したものとみなします。
```

同様に環境変数 ROWS(94 頁)に任意文字列が指定されていない場合は分割されません。次のようなメッセージが表示され、ROWID_SPLIT_MIN_SIZE には 0 が指定されたものとみなします。

```
Warning - 環境変数 ROWS に値が設定されていない場合は、環境変数 ROWID_SPLIT_MIN_SIZE
へ指定した値は無視され 0 バイトを指定したものとみなします。
```

出力されるデータ・ファイルの名前は環境変数 PARTITIONING(89 頁)の 2 ビット目の値に応じて次のように変化します。

- ヒープ構成表名が例えば「X」だった場合

PARTITIONING の 2 ビット目の値	分割しない場合のデータファイル名	分割した場合のデータファイル名
0	X.dat	X_nnnnnn.dat
1	X_mm.dat	X_mm_nnnnnn .dat

“mm”は 01 を最小値、16 か PARALLELISM(90 頁)に指定した値のいずれか小さいほう最大値とします。ROWID ベースの分割範囲の順番と mm のとる値には関係性がなく実行の都度、組み合わせが変化することがあります。つまり各分割範囲とそれに関連して生じるデータファイルに名前を付けて関連させる手段は用意されていませんのでご了承ください。

“nnnnnn”は 000001 から 999999 までを範囲とする数値文字列です。実行中に環境変数 DFILE_ALT_DIRS(77 頁)、FILESIZE(82 頁)へそれぞれ設定したイベント条件が発生すると1つずつ増加した値が割り振られます。

制御ファイルは常に「X.ctl」となりますが、LOAD 句の INFILE オプションの引数は出力したデータファイルの名前が記述されます。

21. 集合化クエリ

集合化クエリは、Query 機能を性能面で強化します。環境変数 QUERYFIX(91 頁)、QUERYVAR(92 頁)の指し示す場所に置かれたファイルが次の例ように”半角セミコロン文字(黄色網掛部分)で各問い合わせを区切る”という単純なルールに沿って記述するだけで、区切り文字で区切られたそれぞれの要素問い合わせを指定された並列度::PARALLELISM(90 頁)数を上限に並列実行することができます。

```
select * from emp where job = 'ANALYST'
   and hiredate > to_date(:b_hiredate, 'yyyy-mm-dd');
select * from emp where job = 'CLERK'
   and hiredate > to_date(:b_hiredate, 'yyyy-mm-dd');
select * from emp where job = 'MANAGER'
   and hiredate > to_date(:b_hiredate, 'yyyy-mm-dd');
select * from emp where job = 'PRESIDENT'
   and hiredate > to_date(:b_hiredate, 'yyyy-mm-dd');
select * from emp where job = 'SALESMAN'
   and hiredate > to_date(:b_hiredate, 'yyyy-mm-dd');
```

出力されるデータ・ファイルの名前は環境変数 PARTITIONING(89 頁)の1ビット目の値に応じて次のように変化します。

- 問合せファイル名が例えば「X.sql」だった場合

PARTITIONING の1ビット目の値	分割しない場合のデータファイル名	分割した場合のデータファイル名
0	X.dat	X_nnnnnn.dat
1	X_mm.dat	X_mm_nnnnnn .dat

“mm”は 01 を最小値、問合せファイル中に含まれる要素問合せの数を2桁の数値文字列に置き換えたものを最大値とします。従って集合化クエリでは 100 以上の要素問合せがあるとエラーとなります。要素問合せが問合せファイル中に記述された順番と mm のとる値には関係性がなく実行の都度、組み合わせが変化することがあります。つまり各要素問合せとそれに関連して生じるデータファイルに名前を付けて関連させる手段は用意されていませんのでご了承ください。

“nnnnnn”は 000001 から 999999 までを範囲とする数値文字列です。実行中に環境変数 DFILE_ALT_DIRS(77 頁)、FILESIZE(82 頁)へそれぞれ設定したイベント条件が発生すると1つずつ増加した値が割り振られます。

制御ファイルは常に「X.ctl」となりますが、LOAD 句の INFILE オプションの引数は出力したデータファイルの名前が記述されます。

要素問合せの選択列リスト(select ….from の”…”部分)は、データ型と列順序が同一になるように記述することを薦めます。内部処理的にデータ型と列順序が非同一であっても製品は動作しますがが、出力されるデータの列値に影響を与えます。

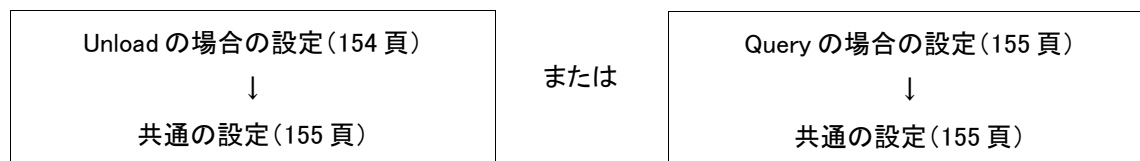
また、制御ファイルの出力時にフィールド名として用いられるのは、必ずしも最初に記述されている要素問合せに由来するものとは限りません。AS 句などで列名を修飾している場合はすべての要素問合せで同じ AS 句を設定してください。

集合化クエリは、サーバーからクライアント間の大量データ移動で待ち時間の多い場合に試してみる価値があります。従いまして、スローダウンが問合せ自身の実行計画や、索引の作り忘れ、CBO 統計情報の更新などの問題に起因している場合や、ボトルネックの原因が Oracle Database サーバーの CPU バウンドや I/O busy にある為、効果を期待できないことがあります。

22. メインフレーム向けデータ移行

この機能は、Oracle Database へ格納されたデータをメインフレームで利用したいというニーズに応えるものです。従来では Pro*COBOL を使って個別にアンロードユーティリティーを開発する必要がありましたが、その開発の手間を MTU が利用する幾つかの環境変数を設定するだけの単純作業に置き換えることができます。

MTU は下記の環境変数を組み合わせて使用することにより SAM (Sequential Access Method；順編成法)と呼ばれている形式のファイル出力を得ることができます。

環境変数の設定手順は下記のように行ってください。

SAM 形式ファイルは BCD や UTF-16BE のマルチバイト文字を含んでいることがあるので、Windows に付属のメモ帳のような普通のテキストエディタでは、文字化けが生じて出力結果のデータファイルを確認することができません。確認が必要な場合には 16 進数のダンプツールかバイナリ編集が可能な高機能エディタをご用意ください。

22.1. Unload の場合の設定

Unload を実行する場合のみこの手順を実施してください。Unload は出力対象に選ばれた表のフルセット・データを戻します。

選択リストや述語をもつ SELECT 文、結合文、集計演算子を持つ問い合わせ、シノニムやビュー、MVIEW、リモート表に対する問い合わせ結果を戻したい場合は Unload ではなく Query をお使いください。

1. ＜必須＞環境変数 LISTTABLE（84 頁）の値を設定する

   出力対象の表の名前を記述してください。半角カンマ文字で区切ることで複数の表を対象に選ぶことができます。何も指定しない場合は省略時初期値の”%”が設定され、これは SRC_USER（97 頁）の所有するテーブルすべてを意味します。

2. ＜オプション＞環境変数 FILETABLE（82 頁）の指し示すファイルの内容を編集する

   出力対象表の名前をテキストファイルを使って指定することができます。LISTTABLE の選択対象とファイルから読み取った表の名前の和集合が最終的に出力対象として選択されます。

3. ＜必須＞環境変数 LISTFIXED（84 頁）の値を設定する

   SAM ファイルは固定長形式なのでこの環境変数を使用します。LISTTABLE と FILETABLE で指定された表の内どれを固定長形式で出力するかを選ぶ機能があるのでワイルドカード文字の”%%”を指定します。パーセント記号が2つ必要なのは、Windows によって環境変数の引用として扱われてしまうのをエスケープするためです。

4. ＜オプション＞環境変数 FILEFIXED（81 頁）の指し示すファイルの内容を編集する

   固定長出力対象の表名をファイルから与えたい場合は、この環境変数を使用します。

22.2. Query の場合の設定

Query を実行する場合のみこの手順を実施してｊください。

1. ＜必須＞環境変数 QUERYFIX（91 頁）の値を設定する

   この変数に指定されたディレクトリ名、ファイル名、SELECT 文はいずれも固定長形式の出力結果を戻します。SAM 形式ファイルは固定長なのでこの値の設定が必要となります。

22.3. 共通の設定

Unload、Query いずれの機能を使用する場合でもこの手順は実施してください。

1. ＜必須＞環境変数 ROWSEPARATOR（95 頁）の値へ”0x00”を設定する

   この設定により行と行の間にセパレータのないデータ形式で出力されます。

2. ＜必須＞環境変数 NUMBER2BCD（88 頁）の値を設定する

   この設定により Oracle Database の NUMBER 型は BCD へ変換されます。

3. ＜オプション＞環境変数 CHARSETFORM（75 頁）の値へ”2”を設定する

   この設定により、人名、住所、商品名などマルチバイト文字の含まれるカラムが NCHAR や NVARCHAR2 として定義されている場合、各国語データベースキャラクタセット（AL16UTF16）で出力されます。

4. ＜オプション＞環境変数 REVERSEBYTEORDER（93 頁）の値に任意文字列（”true”など）を設定する

   CHARSETFORM=2 の設定と組み合わせて使用します。マルチバイト文字のバイト順序を反転します。Intel 系プロセッサが Little Endian なのに対して IBM 互換のメインフレーム用プロセッサでは Big Endian ですのでデータの互換性を得るために変換が必要となります。

23. LOB 出力機能について

v4.18 から LOB を出力する方法が2通り選べるようになりました。このセクションではそれぞれの方法について解説します。

23.1. LOBFILE 方式

こちらは以前のバージョンでも使用可能であった方式です。内部 LOB を個々の値ごとに個別のファイル(=LOBFILE)として出力します。LOB 毎に個別のスレッドを割当てて並列処理を行うので、LOB の平均サイズが大きい場合(数メガバイト以上)で、かつプラットフォームの I/O スケーラビリティーに問題がなければ、1 以上の並列度で能率的に出力することが可能になります。

割当てられるスレッドの数は環境変数 NUM_ASYNCLOBS(87 頁)で指定することができます。出荷時初期値では環境変数 PARALLELISM(90 頁)と同じ値が設定されています。

23.1.1. データ表現

内部のデータ表現はそれぞれ下記の通りとなります。

データ型	データ表現	BOM
CLOB	環境変数 NLS_LANG の第3セグメントか、環境変数 CHARSETID(75 頁) 指定したキャラクタセット ID に依存。両方とも指定された場合は CHARSETID が優先されます。	なし
NCLOB	UTF-16BE 固定	
BLOB	先頭の 4 バイトは、LOB 値本体ではなく LOB のバイト長を示します。バイナリの符号無し 4 バイト整数(バイト順序はリトル・エンディアン)の後方に LOB 値本体が続きます。	N/A

LOB が Empty の場合は 0 バイトのファイルが出力されます。LOB が IS NULL の場合は LOBFILE は作られません。

23.1.2. ファイル命名ルール

LOBFILE は次の規則で命名されます。

- 環境変数 PARTITIONING(89 頁)の選択で「単一ファイル出力」を選んだ場合

```
<表名>_R9999999999_C9999.<拡張子>
```

- 「単一ファイル出力」を選ばなかった場合

```
<表名>_<パーティション名>_R9999999999_C9999.<拡張子>
```

拡張子は CLOB/NCLOB の場合、環境変数 EXTNAMECLOB(79 頁)、BLOB の場合、環境変数 EXTNAMEBLOB(79 頁)に指定された値がそれぞれ選択されます。半角数値の 9 の並びは 1 をスタートとする左側ゼ

ロ詰めの 10 進数が置かれます。R は行番号、C は列番号を示します。

23.1.3. プライマリ・データ・ファイル

すべての LOBFILE はプライマリ・データ・ファイル（PDF,拡張子 dat）から参照され、LOB 値以外の他の行データと関連付けすることができます。LOB 以外のデータ型はプライマリ・データ・ファイルにデータが出力されますが、LOB 値に対応するフィールドには LOB 値の代わりに LOBFILE の名前が出力されます。名前のルールについては前項を参照して下さい。LOB が IS NULL の場合は'NONE'という文字列が LOBFILE 名の代わりに出力されます。

下記に PDF の例を示します。

```
00000000701,~~TEST31_R0000000001_C0002.clo~~,~~TEST31_R0000000001_C0003.clo~~,
00000000463,~~TEST31_R0000000002_C0002.clo~~,~~NONE~~,
00000000464,~~TEST31_R0000000003_C0002.clo~~,~~NONE~~,
00000000707,~~TEST31_R0000000004_C0002.clo~~,~~TEST31_R0000000004_C0003.clo~~,
00000000706,~~TEST31_R0000000005_C0002.clo~~,~~TEST31_R0000000005_C0003.clo~~,
00000000462,~~NONE~~,~~TEST31_R0000000006_C0003.clo~~,
00000000705,~~TEST31_R0000000007_C0002.clo~~,~~TEST31_R0000000007_C0003.clo~~,
00000000468,~~NONE~~,~~TEST31_R0000000008_C0003.clo~~,
00000000469,~~TEST31_R0000000009_C0002.clo~~,~~NONE~~,
000000004710,~~NONE~~,~~TEST31_R0000000010_C0003.clo~~,
```

23.1.4. 制御ファイル

LOBFILE 方式の LOB ファイルを正しくリロードできるよう SQL*Loader 用の制御ファイルも書式が整えられて出力されます。下記に例を示します。

```
LOAD INFILE "TEST31" "VAR 10"
INTO TABLE "TEST31"
TRUNCATE REENABLE FIELDS TERMINATED BY ','
(  "INTEGER_DATA" DECIMAL EXTERNAL(11)
,  CLOB_DATA_LOB  FILLER CHAR(28) ENCLOSED BY '~~'
,  "CLOB_DATA"    LOBFILE (CLOB_DATA_LOB) TERMINATED BY EOF
                  NULLIF CLOB_DATA_LOB = 'NONE'
,  NCLOB_DATA_LOB FILLER CHAR(28) ENCLOSED BY '~~'
,  "NCLOB_DATA"   LOBFILE (NCLOB_DATA_LOB CHARACTERSET AL16UTF16
                  BYTEORDER BIG BYTEORDERMARK NOCHECK) TERMINATED BY EOF
                  NULLIF NCLOB_DATA_LOB = 'NONE'
)
```

23.1.5. 制限

LOBFILE 方式でアンロード可能な LOB の大きさの上限は次の通りです。

データ型	Oracle Database クライアントのバージョン	
	10g 未満	10g 以上

CLOB	符号無し4バイト整数の最大値 単位：バイト	符号無し8バイト整数の最大値 単位：バイト
NCLOB		
BLOB		

23.2. SDF 方式

セカンダリ・データ・ファイル（SDF）方式は、v4.18 からサポートされました。環境変数 MERGE_LOBS_INTO_SDF（85 頁）に値を設定すると利用することができます。SDF 方式は LOBFILE 方式と比較するとファイルの作成される数を少なく抑えることができるので、個数が非常に多い[30]LOB を移行させる場合にカーネル・メモリや所要時間短縮の効果があります。

更に Oracle Database クライアント及び接続する先の Oracle Database が共に 11g 以上である場合、「LOB プリフェッチ」という機能が自動的に有効化され、アンロード性能が飛躍的に向上します。

その反面、使用にあたり、次の制約があります。

- Oracle Database クライアントが 10g 以上であること。
- 環境変数 PARTITIONING（89 頁）を 0 にして使用する事。

23.2.1. データ表現

内部のデータ表現はそれぞれ下記の通りとなります。

データ型	データ表現	BOM
CLOB NCLOB	先頭の 10 文字を使って、それより後に続く LOB 値本体の文字数を'0'から'9'までの文字の並びを使用した 10 進数で表現します。バイト数ではない事に注意してください。 この文字数フィールド及び LOB 値本体は、UTF-16BE でエンコーディングされます。文字数フィールドとそれに続く LOB 値本体が対を形成し、この対が同じ LOB 列に含まれる LOB の数だけ繰り返されます。繰り返す順序はプライマリ・データ・ファイル（PDF）へ出力された行データの順序と同じです。	環境変数 NOBYTEORDERMARK（87 頁）の設定に依存
BLOB	LOB 値のバイト長を符号無し 4 バイト整数（リトル・エンディアン）で表現したフィールドとそれより後に続く LOB 値本体が対を形成し、この対が同じ LOB 列に含まれる LOB の数だけ繰り返されます。順序については上記と同じです。	N/A

LOB が Empty 及び IS NULL の場合は CLOB/NCLOB では 10 文字分の'0'が出力されます（LOB 値は無し）。また BLOB では符号無し 4 バイト整数の 0 が出力されます。

23.2.2. ファイル命名ルール

SDF は次の規則で命名されます。LOBFILE 方式とは異なり行数のラベルがない事に注意して下さい。

```
<表名>_C9999.<拡張子>
```

[30] 10 万個以上が"多い"と判断する基準の一例です。

拡張子は CLOB/NCLOB の場合、環境変数 EXTNAMECLOB（79 頁）、BLOB の場合、環境変数 EXTNAMEBLOB（79 頁）に指定された値がそれぞれ選択されます。半角数値の 9 の並びは 1 をスタートとする左側ゼロ詰めの 10 進数が置かれます。C は列番号を示します。

23.2.3. プライマリ・データ・ファイル

SDF 上の個々の LOB 値はプライマリ・データ・ファイル（PDF,拡張子 dat）から参照され、LOB 値以外の他の行データと関連付けすることができます。PDF に並ぶ行の順序は入れ替えないようご注意ください。この順序は SDF 中に並ぶ LOB の順序と一致します。

LOB 以外のデータ型はプライマリ・データ・ファイルにデータが出力されますが、LOB 値に対応するフィールド部分には LOB 値の代わりに LOB であることを示す文字列定数が出力されます。LOB が IS NULL の場合は'NONE'という文字列が出力されます。LOB が IS NOT NULL の場合は'have'という文字列が出力されます。従いまして LOB が IS NULL であることと Empty であることの区別は、SDF 上の文字数（or バイト長）フィールドと PDF 上の NONE という文字列を組み合わせて判断します。

下記に PDF の例を示します。

```
00000000221,~~have~~,~~have~~,
00000000223,~~have~~,~~NONE~~,
00000000224,~~have~~,~~NONE~~,
00000000227,~~have~~,~~have~~,
00000000226,~~have~~,~~have~~,
00000000229,~~have~~,~~NONE~~,
000000002310,~~NONE~~,~~have~~,
00000000222,~~NONE~~,~~have~~,
00000000225,~~have~~,~~have~~,
00000000228,~~NONE~~,~~have~~,
```

23.2.4. 制御ファイル

SDF 方式の LOB ファイルを正しくリロードできるよう SQL*Loader 用の制御ファイルも書式が整えられて出力されます。下記に例を示します。

```
LOAD INFILE "TEST31" "VAR 10"
INTO TABLE "TEST31"
TRUNCATE REENABLE FIELDS TERMINATED BY ','
(  "INTEGER_DATA" DECIMAL EXTERNAL(11)
,  CLOB_DATA_LOB  FILLER CHAR(4) ENCLOSED BY '~~'
,  "CLOB_DATA"    LOBFILE (CONSTANT 'TEST31_C0002.clo' CHARACTERSET AL16UTF16
                  BYTEORDER BIG BYTEORDERMARK CHECK) VARCHARC(10, 1000)
                  NULLIF CLOB_DATA_LOB = 'NONE'
,  NCLOB_DATA_LOB FILLER CHAR(4) ENCLOSED BY '~~'
,  "NCLOB_DATA"   LOBFILE (CONSTANT 'TEST31_C0003.clo' CHARACTERSET AL16UTF16
                  BYTEORDER BIG BYTEORDERMARK CHECK) VARCHARC(10, 1000)
                  NULLIF NCLOB_DATA_LOB = 'NONE'
```

```
)
```

23.2.5. 制限

SDF 方式でアンロード可能な LOB の大きさの上限は次の通りです。

データ型	最大データ長
CLOB	9999999999 文字
NCLOB	
BLOB	符号無し 4 バイト整数の最大値、単位：バイト

24. 性能評価方法

プラットホーム OS が提供している、性能診断ツールを使用することにより製品の性能特性を客観的に評価することが可能です。この章では、パフォーマンスモニタ、及びデータ・コレクター・セット呼ばれる Windows の性能診断ツールを使う方法について解説します。

24.1. パフォーマンスモニタ

パフォーマンスモニタは MMC プラグインのひとつとして提供されています。パフォーマンスモニタは GUI を持つので、性能特性を視覚的に手軽にチェックするのに向いています。

ただし、表示可能な対象統計期間がグラフ上の狭い範囲に限られてしまうので、長期的な性能を観測するのには適していません。また性能統計情報をオフラインで検証する事も出来ません。このような用途には次のセクションに説明した「24.2.データ・コレクター・セット（164 頁）」が適しています。

初めにパフォーマンスモニタを使用する前に下記のコマンドを実行してプラグインを組込む対象の Windows 管理コンソールを起動します。

```
MMC
```

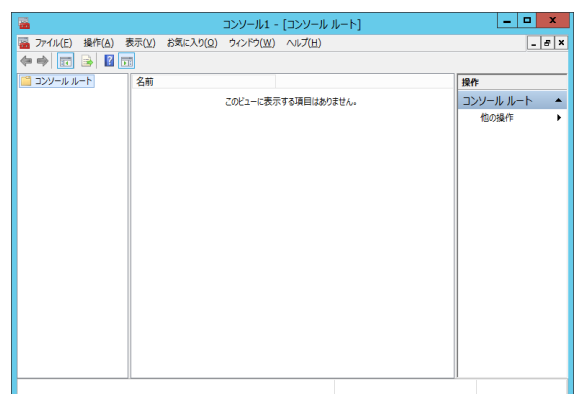
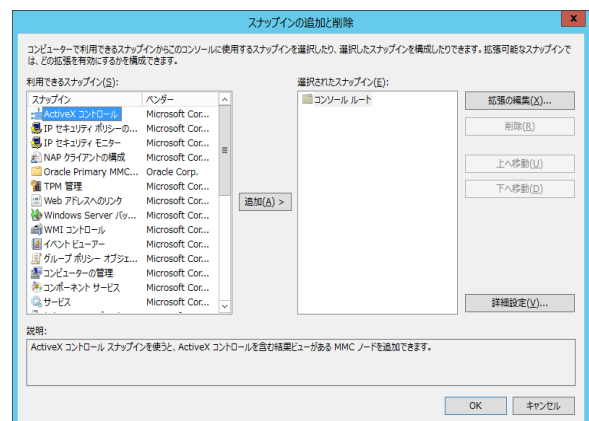
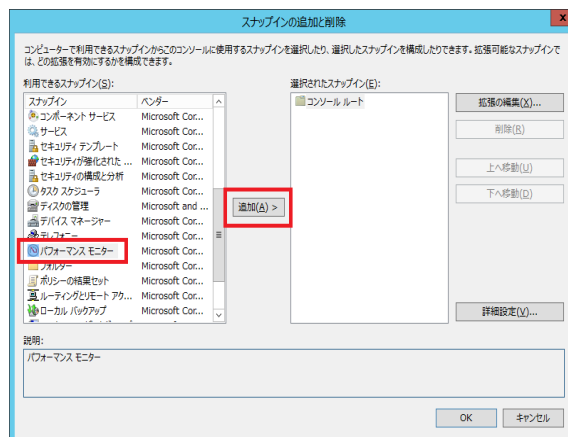
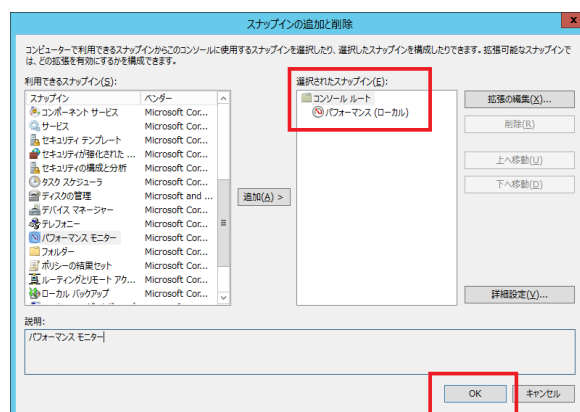
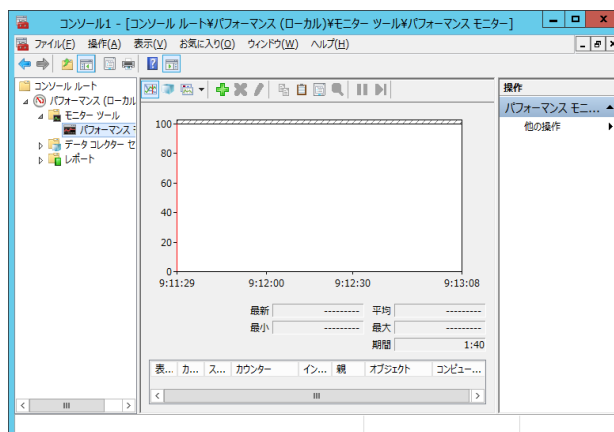
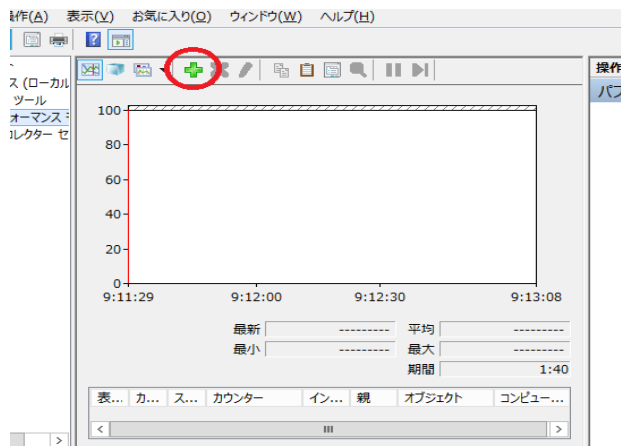
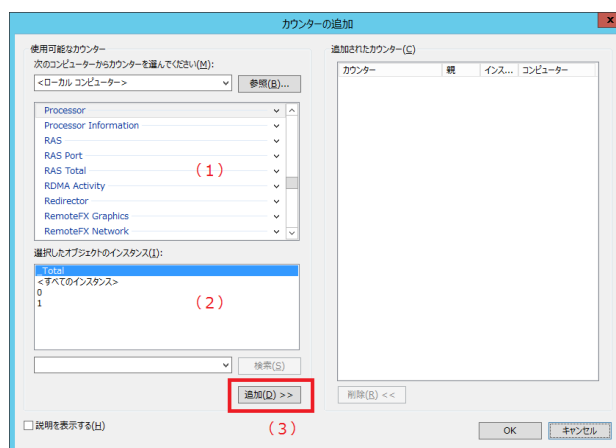
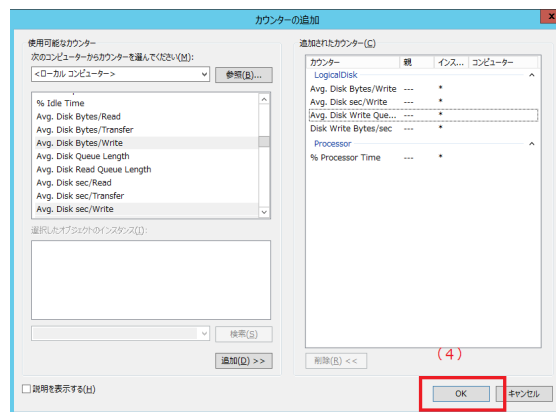
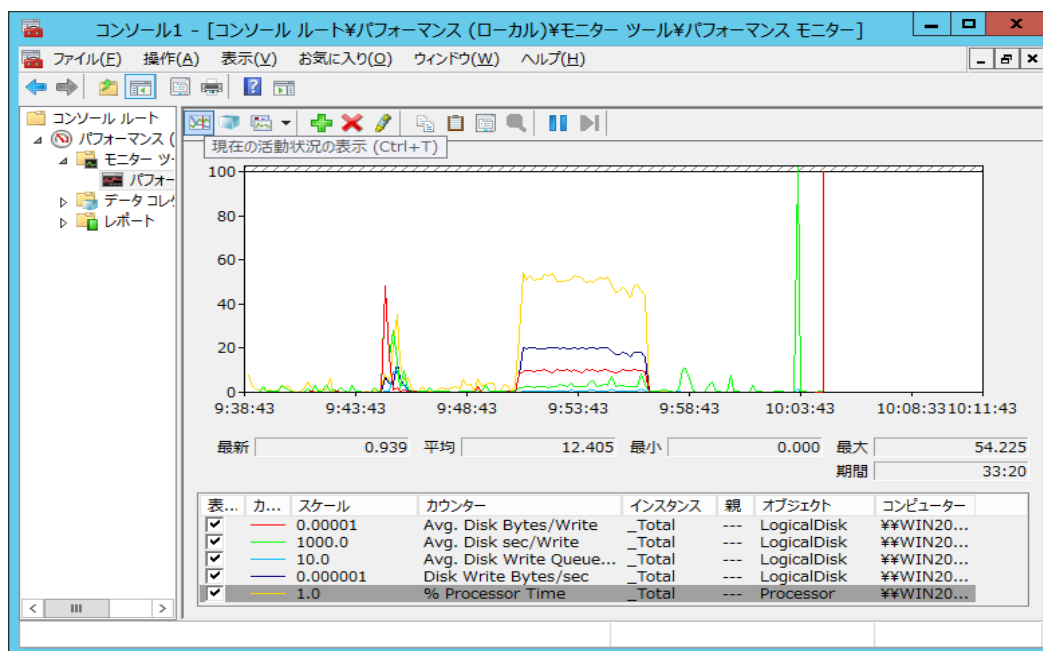
コマンドの実行により右図の画面が表示されます。

次にメニューから[ファイル]→[スナップインの追加と削除]を選びます。右図のダイアログボックスが表示されます。

左側の[利用できるスナップイン]リストボックスから[パフォーマンスモニタ]を選択し、[追加]ボタンを押します。

[選択されたスナップイン]へ[パフォーマンスモニタ]が表示されます。OK ボタンを押してダイアログボックスを閉じます。

左側のアウトラインから[パフォーマンス]を選択し、サブフォルダを展開します。さらに[モニターツール]→[パフォーマンスモニター]を選択します。

中央のパネルのツールバーから[＋]記号ボタンを押します。

評価に使用するカウンターを選択し[追加]ボタンを押します。選択されたカウンタが[追加されたカウンター]に表示されます。

製品の性能特性評価に有用と思われるカウンターを次の表に示します。

表 31:推奨パフォーマンスカウンター

カウンター	スケール	色	上限値	摘要
カテゴリ:LogicalDisk				
Avg Disk bytes/Write	0.00001	赤	10MB/回	OVERLAP_BUFFER_LENGTH(89 頁)に指定した値の影響が強く現れます。
Avg Disk sec/Write	1000	緑	100 ミリ秒/回	大きな書込みサイズでは Avg Disk bytes/Write との関係が正比例になります。小さな書込みサイズでの性能特性を見るのに適したカウンタです。
Avg Disk Write Queue Length	10	青	10 個	並列度の設定が適切かどうかを判断する指標です。データファイルを書き込むストレージのスケーラビリティに余裕がある場合、並列度よりも小さい値を示します。
Disk Write Bytes/sec	0.000001	藍	100MB/秒	データ率を示します。この値が大きければ大きいほど高速に動作しているといえます。プログラムを実行した時にコンソールに表示される統計量の内「処理速度(kbytes/秒)」とほぼ同じ値が示されます。 処理能力のプライマリ・インジケータです。
カテゴリ:Processor				
% Processor Time	1.0	黄	100%	CPU バウンドかそうでないかを判断するのに使います。

性能特性評価に使用するカウンタをすべて[追加されたカウンター]にリストアップしたら[OK]ボタンを押します。

上記の設定した場合の、特性曲線の例を下に示します。

24.2. データ・コレクター・セット

データ・コレクター・セットは、オフラインでの性能データ分析や長期間に及ぶ観測、コンピュータ上で発生する他のイベントによる影響を調査するのに役立ちます。

使用するパフォーマンス・カウンタを集めたデータ・コレクター・セットは、logman コマンドを使って作成／削除／制御する事が出来ます。

データ・コレクタ・セットを使って性能データを収集するにはあらかじめ"Performance Logs & Alerts"サービスを開始しておく必要があります。Windows では起動後のこのサービスは停止状態にあります。

```
net start "Performance Logs & Alerts"
```

24.2.1. 作成

たとえば MTU の性能特性を検証するには次のようなコマンドを実行して「mtu_parf_log」という、データ・コレクター・セットを作成して下さい。

```
logman create counter mtu_parf_log -c "¥LogicalDisk¥Avg Disk bytes/Write"
"¥LogicalDisk¥Avg Disk sec/Write" "¥LogicalDisk¥Avg Disk Write Queue Length"
"¥LogicalDisk¥Disk Write Bytes/sec" "¥Processor(_Total)¥% Processor Time" -si
00:01:00 -sc 1440 -b 2013/7/5 15:50
```

-c オプションを付けて選択したパフォーマンス・カウンタは「表 31:推奨パフォーマンスカウンター(163 頁)」に示したものと同じです。もちろん、これ以外にも選択可能です。

-si, -sc, -b に指定したオプションの値は「2313 年 7 月 5 日の 15 時 50 分から開始して、1 分おきに 1440 回サンプリングを継続する」という意味になります。

データ・コレクター・セットの登録が正常終了すると次のようなメッセージが表示されます。

```
コマンドは、正しく完了しました。
```

次のようなメッセージが現れ失敗する場合は「管理者として実行」にてコマンドプロンプトを開き再実行して下さい。

```
エラー:アクセスが拒否されました。
制限されているトークンで実行しています。管理者特権を使用して実行してみてください。
```

登録されたデータ・コレクタ・セットの状態は query アクションで確認する事が出来ます。例を示します。

```
C:¥Windows¥system32>logman query mtu_parf_log

名前:                    mtu_parf_log
状態:                    停止
ルート パス:                %systemdrive%¥PerfLogs¥Admin
セグメント:                 オフ
スケジュール:                オン
別のユーザーとして実行:           SYSTEM

スケジュール
開始日:                   7/5/2013
開始時刻:                   3:50:00 PM

名前:                    mtu_parf_log¥mtu_parf_log
種類:                    カウンター
追加:                    オフ
循環:                    オフ
上書き:                    オフ
サンプルの間隔:                60 秒

カウンター:
  ¥LogicalDisk¥Avg Disk bytes/Write
  ¥LogicalDisk¥Avg Disk sec/Write
  ¥LogicalDisk¥Avg Disk Write Queue Length
  ¥LogicalDisk¥Disk Write Bytes/sec
  ¥Processor(_Total)¥% Processor Time
```

```
コマンドは、正しく完了しました。
```

24.2.2. 開始

「状態」が「停止」となっているデータ・コレクター・セットで、性能データの収集を開始するには start アクションを指定して下さい。次に例を示します。

```
logman start mtu_parf_log
```

性能データは「ルートパス」の下層へ「名前」_<6 桁数字>.blg の書式を持つログ・ファイルへ保存されます。query アクションを使うと出力場所のパスを確認する事が出来ます。

```
C:¥Windows¥system32>logman query mtu_parf_log

名前:                    mtu_parf_log
状態:                    実行中
ルート パス:                %systemdrive%¥PerfLogs¥Admin
セグメント:                 オフ
スケジュール:                オン
別のユーザーとして実行:            SYSTEM

スケジュール
開始日:                   7/5/2013
開始時刻:                   3:50:00 PM

名前:                    mtu_parf_log¥mtu_parf_log
種類:                    カウンター
出力場所:                   C:¥PerfLogs¥Admin¥mtu_parf_log_000001.blg
追加:                    オフ
循環:                    オフ
上書き:                    オフ
サンプルの間隔:                 60 秒

カウンター:
  ¥LogicalDisk¥Avg Disk bytes/Write
  ¥LogicalDisk¥Avg Disk sec/Write
  ¥LogicalDisk¥Avg Disk Write Queue Length
  ¥LogicalDisk¥Disk Write Bytes/sec
  ¥Processor(_Total)¥% Processor Time

コマンドは、正しく完了しました。
```

このログは、データ・コレクタ・セットが停止するか、次のログに切り替わった後に、パフォーマンス・モニタから開いて内容を GUI から確認する事が出来ます。

24.2.3. 停止

「状態」が「実行中」となっているデータ・コレクタ・セットで、性能データの収集を停止するには stop アクションを指定して下さい。次に例を示します。

```
logman stop mtu_parf_log
```

24.2.4. 削除

データ･コレクタを削除するには delete アクションを指定して下さい。次に例を示します。

```
logman delete mtu_parf_log
```

25. プログラムの引数

製品プログラムは動作条件を引数として与える事ができます。MTU に定義されている引数には下記のものがあります。

表 32 プログラムの引数

位置		意味

1	必須	接続する先のユーザ名、パスワード、およびオプションでデータベース別名を与えます。書式[31]は下記の通りとなります。「8.8.接続先の指定(47頁)」に具体例があるので併せてご確認下さい。
		`<uUsername>/<password>[@db_alias]`
		但し、Oracle 10g 以降の Client または Instant Client をお使いの場合「簡易接続ネーミング・メソッド」と呼ばれている下記の書式も受入れる事ができます。簡易接続ネーミング・メソッドを使用する際は環境変数 TNS_ADMIN に値が設定されていないことをご確認ください。
		`<uUsername>/<password>[@[//]host[:port][/service_name]]`
		password 部分に1文字分の半角アスタリスクを設定すると、パスワード入力用のダイアログボックス(「図 11 パスワード入力のダイアログボックス(MTU)(168 頁)」参照)が開き、パスワードを実行時に入力する事ができます。入力されたパスワードは文字数分のアスタリスク文字がエコーバックされるので、入力を覗き見されたとしてもパスワードの漏えいを防ぐ事ができます。
		上記の引数を平文で設定する代わりにパスワードの秘匿化機能を使用することも出来ます。詳しくは「15. パスワードの秘匿化について 128頁)」にある記事を参照してください。
		接続先 DB が Oracle 11g 以上で、かつ初期化パラメータ SEC_CASE_SENSITIVE_LOGON が TRUE である場合、パスワードの大文字／小文字が区別されますのでこの引数の指定時にはご注意下さい。

2	必須	「表 16 各機能と出力される表題(101頁)」の“機能”欄に表示した文字列を指定します。それぞれの説明については同表をご確認下さい。

図 11 パスワード入力のダイアログボックス(MTU)

引数を何も指定せずに製品プログラムを実行すると起動方法の簡単な説明が表示されて終了します。下記に表示例を示します。

[31] <> で囲まれた要素は必須入力です。[]で囲まれた要素は任意入力です。

```
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
Windows 7 Professional Service Pack 1 (build 7601), 64-bit native
Number of NUMA nodes: 1
Number of physical processor packages: 1
Number of processor cores: 2
Number of logical processors: 2
Number of processor L1/L2/L3 caches: 4/1/0
Error - 引数が不足しています。
コマンド書式 : MTU [userid] [func]
userid : 必須。接続先ユーザ名/パスワード@接続文字列
  例：scott/tiger@orcl.world （接続文字列省略時はIPC接続）
  例：/@orcl.world （OS認証ではユーザ名/パスワードを省略可能）
func   : 必須。下記機能の何れか一つを指定。
  install     : 製品コンポーネントのインストール
  unload      : 表データファイル出力
  query       : 任意問合せファイル出力
  getmeta     : スキーマオブジェクト定義SQL文出力
  consdisable : 整合性制約の無効化
  consenable  : 整合性制約の有効化
  getfkrb     : 外部キー再構成スクリプト出力
  gettblp     : 表間参照階層図（正展開）の出力
  gettbln     : 表間参照階層図（逆展開）の出力
  getplsp     : 名前付きPL/SQL間参照階層図（正展開）の出力
  getplsn     : 名前付きPL/SQL間参照階層図（逆展開）の出力
異常終了しました。(mtu.c-392)
```

26. プログラムの戻り値

製品プログラムは終了時に戻り値を返すことが出来るため、お客様自身が作成されたシェル・スクリプト等へ組み込む場合に、後続する手続きへの分岐条件を与えることが出来ます。以下に戻り値とその意味を示します。

表 33 プログラムの戻り値

戻り値	意味
0	正常終了
1 以上	異常終了

27. トラブル・シューティング

27.1. 製品コンポーネントインストール時のエラーメッセージ

インストール時に発生しやすいエラーの事例と対処法について説明します。

Error – OCI.DLL をロードできません。下記の点を確認して下さい。

```
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
Windows 7 Professional Service Pack 1 (build 7601), 64-bit native
Number of NUMA nodes: 1
Number of physical processor packages: 1
Number of processor cores: 2
Number of logical processors: 2
Number of processor L1/L2/L3 caches: 4/1/0
表データファイル出力を実行します...
Error - Win32API の実行に失敗しました。Code=0x7e, Msg=指定されたモジュールが見つかり
ません。
関数 rcChkWin32Api() が異常終了しました。(rcChkWin32Api.c-46)
Error - OCI.DLL をロードできません。下記の点を確認して下さい。
1, Oracle 製品がマニュアル通りに正しくインストールされていること。
2, 環境変数 PATH へ OCI.DLL を含むディレクトリがエントリされていること。
現在の PATH の値を表示します：
  C:¥Program Files¥PLUMSIX¥mtu_4.2¥
  C:¥Program Files (x86)¥Oracle¥jre¥1.3.1¥bin
  C:¥Program Files (x86)¥Oracle¥jre¥1.1.8¥bin
  C:¥Windows¥system32
  C:¥Windows
  C:¥Windows¥System32¥Wbem
  C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥
関数 rcImportOciLibrary() が異常終了しました。(psOciCommon.c-708)
関数 rcOciSvc() が異常終了しました。(cOciSvc.c-1298)
異常終了しました。(mtu.c-550)
```

このエラーは製品プログラムを実行したコンピュータに Oracle Client または Oracle Server がインストールされていない場合か、環境変数 PATH が正しく oci.dll の場所を指していない場合に発生します。oci.dll は Oracle 製品プログラムに含まれる、重要な動的リンクライブラリのひとつです。

x64 ベースの Windows 上で 32-bit 版の MTU を使用する場合には、32-bit 版の Oracle Client または Oracle Server がインストールされている必要があります。未インストールの場合、このエラーが発生します。

製品プログラムを実行するコンピュータを Oracle Client または Oracle Server がインストールされているものに替えるか、コンピュータへ Oracle Client または Oracle Server をインストールしてから再試行して下さい。

Error – OCIのバージョンが旧すぎます。

このエラーは、Oracle 9i Release 2 未満のクライアントを使用しているときに発生します。

ORA-12154: TNS: 指定された接続識別子を解決できませんでした

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>install
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
製品コンポーネントのインストールを実行します...
ネット・サービス名 xxxx を解決中...
ORA-12154: TNS: 指定された接続識別子を解決できませんでした
Error - ネット・サービス名 xxxx が TNSNAMES.ORA、または Oracle Names へ登録済か確認し
てください。
関数 rcAttachToServer() が異常終了しました。(psOciCommon.c-885)
関数 rcOciSvc() が異常終了しました。(cOciSvc.c-1194)
所要時間（秒）                  :          0.07
異常終了しました。(mtu.c-981)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

「Error - 」以降に表示されたメッセージに有る通り、未定義の Oracle ネット・サービス名を使おうとしています。Env.bat の「set USERID=」ステートメントに指定されたネット・サービス名にスペルミス等のエラーが無いかどうかをチェックしてください。

ORA-12541: TNS: リスナーがありません

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>install
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
製品コンポーネントのインストールを実行します...
ネット・サービス名 xxxx を解決中...
ORA-12541: TNS: リスナーがありません
Error - ネット・サービス名 xxxx に定義された IP アドレスとポート番号の位置に動作中のリ
スナがあるか確認して下さい。
関数 rcAttachToServer() が異常終了しました。(psOciCommon.c-885)
関数 rcOciSvc() が異常終了しました。(cOciSvc.c-1194)
所要時間（秒）                  :          1.18
異常終了しました。(mtu.c-981)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

接続要求先のリスナーが稼動しているか、リスナーが稼動しているコンピュータの IP アドレスとポート番号の指定に誤りが無いかどうかチェックしてください。

ORA-01034: ORACLE not available

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>install
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
製品コンポーネントのインストールを実行します...
ネット・サービス名: xxxx へユーザ名:system として接続します。
しばらくお待ちください...
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
プールを作成します(cOciSes:FIFO)...
ORA-01034: ORACLE not available
```

```
ORA-27101: shared memory realm does not exist
Error - インスタンスが起動されていません。
関数 rcOciSes() が異常終了しました。(cOciSes.c-218)
関数 rcOciSesInPool() が異常終了しました。(cOciSes.c-285)
関数 rcAddOrDelItems() が異常終了しました。(cPool.c-358)
関数 rcPulseOrWaitForEvent() が異常終了しました。(cEvent.c-247)
Information - プールの状態 szItemType=cOciSes, iMaxItems=0, iNumFree=0,
rCtx.iDifference=1
関数 rcChgNumItemsPool() が異常終了しました。(cPool.c-441)
関数 rcPool() が異常終了しました。(cPool.c-526)
関数 rcOciSvc() が異常終了しました。(cOciSvc.c-1327)
所要時間 (秒)                  :         0.20
異常終了しました。(mtu.c-981)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

接続先のオラクルのインスタンスがスタートアップしているかどうか alert.log などを確認してください。

ORA-12560: TNS: プロトコル・アダプタ・エラーが発生しました

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>install
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
製品コンポーネントのインストールを実行します...
IPC 接続を解決中....
ORA-12560: TNS: プロトコル・アダプタ・エラーが発生しました
Error - IPC 接続を利用するときは有効な ORACLE_SID を設定してください。
関数 rcAttachToServer() が異常終了しました。(psOciCommon.c-885)
関数 rcOciSvc() が異常終了しました。(cOciSvc.c-1194)
所要時間 (秒)                  :         0.06
異常終了しました。(mtu.c-988)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

IPC[32]によるローカル接続を要求しようとしましたが、接続先の Oracle インスタンスを識別することが出来ませんでした。「Error - 」以降に表示されたメッセージに有る通り、環境変数 ORACLE_SID か、あるいは HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥<key_name>キーへ ORACLE_SID を明示的に定義し接続先インスタンス名を指定してください。

ORA-01017: ユーザー名/パスワードが無効です。ログオンは拒否されました。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>install
Multi-threaded Unloader version 4.2
(c) 2003 Plumsix Co.,Ltd. All rights reserved.
製品コンポーネントのインストールを実行します...
ネット・サービス名: xxxx へユーザ名:systen として接続します。
しばらくお待ちください...
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
プールを作成します(cOciSes:FIFO)...
ORA-01017: ユーザー名/パスワードが無効です。ログオンは拒否されました。
```

[32] IPC はプロセス間通信(Interprocess Communication)の略称。リスナーと同じノードに存在するクライアント・アプリケーションが、サーバー・プロセスとの通信に使用するプロトコルの 1 つ。一般的に IPC は、TCP/IP よりデータ転送速度に優れています。

```
Error - ユーザ名 systen が存在するか、あるいはパスワードの有効性を確認して下さい。
関数 rcOciSes() が異常終了しました。(cOciSes.c-218)
関数 rcOciSesInPool() が異常終了しました。(cOciSes.c-285)
関数 rcAddOrDelItems() が異常終了しました。(cPool.c-358)
関数 rcPulseOrWaitForEvent() が異常終了しました。(cEvent.c-247)
Information - プールの状態 szItemType=cOciSes, iMaxItems=0, iNumFree=0,
rCtx.iDifference=1
関数 rcChgNumItemsPool() が異常終了しました。(cPool.c-441)
関数 rcPool() が異常終了しました。(cPool.c-526)
関数 rcOciSvc() が異常終了しました。(cOciSvc.c-1327)
所要時間 (秒)                 :         0.28
異常終了しました。(mtu.c-981)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

Env.bat の「set USERID=」ステートメントに指定されたユーザ名あるいはパスワードにスペルミス等のエラーが無いかどうかをチェックしてください。特に 11g 以降のデータベースでは、ユーザ名とパスワードに含まれるの大文字／小文字が区別されます。

ORA-00955: すでに使用されているオブジェクト名です。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>install
Multi-threaded Unloader version 4.2
                   :                                   :
                   「12.1.表示項目の共通部分(100頁」参照
                   :                                   :
APLDDT01 表作成               :    Executing
ORA-00955: すでに使用されているオブジェクト名です。
関数 rcStmtExecute() が異常終了しました。(cOciStmt.c-540)
関数 rcExecDefineAndFetch() が異常終了しました。(cOciStmt.c-1054)
関数 rcOciStmtVa() が異常終了しました。(cOciStmt.c-1406)
関数 rcOciStmtList() が異常終了しました。(cOciStmt.c-1474)
関数 rcInstall() が異常終了しました。(rcInstall.c-559)
所要時間 (秒)                 :         1.00
異常終了しました。(mtu.c-1004)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

このエラーは製品コンポーネント(表やパッケージ等)の一部が不完全にインストールされているか、または全部が既にインストールされている場合に発生します。

不完全にインストールされている場合は「10.2..第2の方法(選択)(68頁」」に示した手順に沿って、オブジェクトを一旦削除してから再度 Install を試みて下さい。

ORA-01031: 権限が不足しています。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>install
Multi-threaded Unloader version 4.2
                   :                                   :
                   「12.1.表示項目の共通部分(100頁」参照
                   :                                   :
APLDDT01 表作成               :    Executing                            (a)
```

```
ORA-01031: 権限が不足しています。
関数 rcStmtExecute() が異常終了しました。(c0ciStmt.c-540)
関数 rcExecDefineAndFetch() が異常終了しました。(c0ciStmt.c-1054)
関数 rc0ciStmtVa() が異常終了しました。(c0ciStmt.c-1406)
関数 rc0ciStmtList() が異常終了しました。(c0ciStmt.c-1474)
関数 rcInstall() が異常終了しました。(rcInstall.c-559)
所要時間 (秒)                    :          0.26
異常終了しました。(mtu.c-1004)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

このエラーは製品コンポーネント(表やパッケージ等)をスキーマへ作成する為の権限がない場合に発生します。(a)にはエラーになった製品コンポーネント(表やパッケージ)を作る為の SQL 文が表示されるので、そのオブジェクトを作る為に必要なシステム権限が与えられているかどうかを SQL*Plus 等のユーティリティーで確認して下さい。現在割当てられているシステム権限の調べ方は「8. インストール (39頁)」を参照して下さい。

ORA-01950: 表領域' xxxxxx'に対する権限がありません、または ORA-01536: 表領域 xxxxxx に対して割り当てられた領域を使い果たしました。

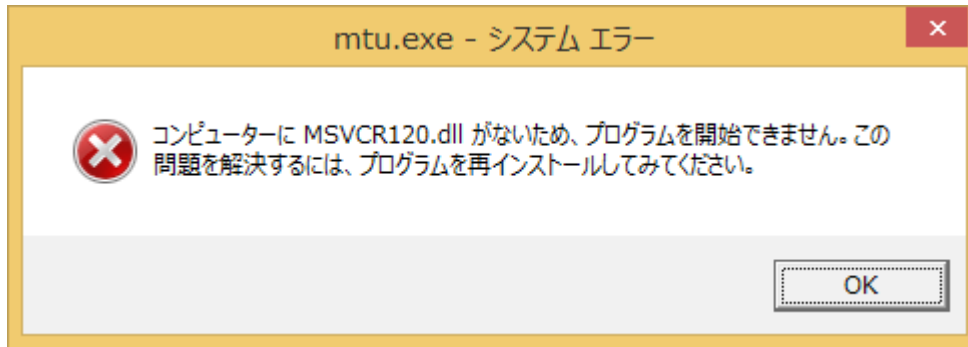
```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>install
Multi-threaded Unloader version 4.2
                    :                                    :
                    「12.1..表示項目の共通部分(100頁」参照
                    :                                    :
APLDDT01 表作成                :    Executing                             (a)
ORA-01950: 表領域'xxxxxx'に対する権限がありません
関数 rcStmtExecute() が異常終了しました。(c0ciStmt.c-540)
関数 rcExecDefineAndFetch() が異常終了しました。(c0ciStmt.c-1054)
関数 rc0ciStmtVa() が異常終了しました。(c0ciStmt.c-1406)
関数 rc0ciStmtList() が異常終了しました。(c0ciStmt.c-1474)
関数 rcInstall() が異常終了しました。(rcInstall.c-559)
所要時間 (秒)                    :          0.34
異常終了しました。(mtu.c-1004)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

このエラーは製品コンポーネント(表)をスキーマへ作成する為の十分なクオータが割り当てられていない場合に発生します。(a)にはエラーになった製品コンポーネント(表)を作る為の SQL 文が表示されるので、接続しているユーザのデフォルト表領域へ表を作る為に必要なクオータが割り当てられているか(あるいはユーザへ UNLIMITED TABLESPACE システム権限が与えられているかどうか)を SQL*Plus 等のユーティリティーで確認して下さい。現在割当てられているクオータの調べ方は「8. インストール (39頁)」を参照して下さい。

27.2. 実行時のエラーメッセージ

実行時に発生しやすいエラーの事例と対処法について説明します。

このコンピュータに MSVCR120.dll がないため、プログラムを開始できません。

MTU が必要とする Microsoft 社の「再頒布可能パッケージ 」がインストールされていません。「6.1.必要システム構成（31 頁）」に入手先に関する案内が示してありますので、「再頒布可能パッケージ 」を先にインストールしてから、MTU の起動を再試行して下さい。

Error – オブジェクトが見つからないか、必要な権限が割り当てられていません。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>getmeta
Multi-threaded Unloader version 4.2
                   :                                   :
                   「12.1..表示項目の共通部分(100頁」参照
                   :                                   :
スキーマオブジェクト定義 SQL 文を出力中です。output¥meta.sql
Error - オブジェクトが見つからないか、必要な権限が割り当てられていません。
所有者名：SCOTT オブジェクト名：APLDDT01 権限：SELECT                    (a)
関数 rcCheckPrivsGiven() が異常終了しました。(cPrivsGiven.c-219)
関数 rcGetMeta() が異常終了しました。(psGetMeta¥rcGetMeta.c-697)
出力バイト総数 (kbytes)         :         1.27
所要時間 (秒)                   :         0.39
処理速度 (kbytes/秒)            :         3.26
異常終了しました。(mtu.c-1004)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

このエラーはインストールが行われていないか、あるいは不完全なまま終わっている場合に発生します。(a)には製品プログラムが動作するために必要なデータベース･オブジェクトの所有者名、名前、アクセスに必要なオブジェクト権限が示されます。「8. インストール (39頁)」を参照の上インストールを完全に実施してください。

Win32API の実行に失敗しました。Code=0x3, Msg=指定されたパスが見つかりません。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>getmeta
Multi-threaded Unloader version 4.2
                   :                                   :
                   「12.1.表示項目の共通部分(100頁」参照
                   :                                   :
対象スキーマは SCOTT です。
Error - Win32API の実行に失敗しました。Code=0x3, Msg=指定されたパスが見つかりません。
関数 rcChkWin32Api() が異常終了しました。(rcChkWin32Api.c-30)
Error - ファイルの名前：outpug¥meta.sql                   (a)
関数 rcFile() が異常終了しました。(cFile.c-408)
関数 rcGetMeta() が異常終了しました。(psGetMeta¥rcGetMeta.c-633)
所要時間 (秒)                   :         0.25
```

```
異常終了しました。(mtu.c-1004)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

このエラーは(a)の部分に示されるファイルの出力先ディレクトリが無効の場合に発生します。

Env.bat 内の環境変数 OUTPUT(89 頁)によって指し示すされるディレクトリが実在するかどうかを確認してください。

Error – Win32API の実行に失敗しました。Code=0x5, Msg=アクセスが拒否されました。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>getmeta
Multi-threaded Unloader version 4.2
                   :                                  :
                   「12.1.表示項目の共通部分(100頁」参照
                   :                                  :
Error - Win32API の実行に失敗しました。Code=0x5, Msg=アクセスが拒否されました。
関数 rcChkWin32Api() が異常終了しました。(rcChkWin32Api.c-30)
Error - ファイルの名前:output¥meta.sql                            (a)
関数 rcFile() が異常終了しました。(cFile.c-408)
関数 rcGetMeta() が異常終了しました。(psGetMeta¥rcGetMeta.c-633)
所要時間 (秒)                :         0.25
異常終了しました。(mtu.c-1004)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

このエラーは(a)の部分に示されるファイルまたはディレクトリの属性が読み込み専用である場合に発生します。属性を読み書き可能に変更してから再試行して下さい。

Error – Win32API の実行に失敗しました。Code=0x70, Msg=ディスクに十分な空き領域がありません。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>unload
Multi-threaded Unloader version 4.2
                   :                                  :
                   「12.1.表示項目の共通部分(100頁」参照
                   :                                  :
表に含まれている行数の測定    :           67:    0.00
整合性制約情報-Z              :          370:    0.02
整合性制約情報-B              :          370:    0.06
出力対象の表名                :           52:    0.94
データ・ファイルと制御ファイルを出力中です。output
TableName                     :       NumRows:  Elapse:Rows/s:Bytes: kB/s:   Len
Error - ファイルの名前:output¥STATS$BG_EVENT_SUMMARY.dat
関数 dwAddNextExtent() が異常終了しました。(cFile.c-74)
Error - Win32API の実行に失敗しました。Code=0x70, Msg=ディスクに十分な空き領域があり
ません。
関数 rcChkWin32Api() が異常終了しました。(rcChkWin32Api.c-30)
関数 rcWriteFile() が異常終了しました。(cFile.c-576)

関数 rcPostFetchAction() が異常終了しました。(psGetData¥cUnLoadUnit.c-3489)
関数 rcSelectIntoVector() が異常終了しました。(cOciStmt.c-802)
関数 rcExecDefineAndFetch() が異常終了しました。(cOciStmt.c-1019)
Waiting.
```

```
Synchronized.POOL_STATISTICS  :    Executing
Error - １つ以上のスレッド関数が異常修了しました。
関数 rcSynchronizeSvc() が異常終了しました。(cOciSvc.c-1414)
関数 rcPostRepGetData() が異常終了しました。(psGetData¥cC1021.c-2016)
Waiting.
Synchronized.
Error - １つ以上のスレッド関数が異常修了しました。
関数 rcSynchronizeSvc() が異常終了しました。(cOciSvc.c-1414)
関数 rcForEachExList() が異常終了しました。(cList.c-397)
関数 rcGetData() が異常終了しました。(psGetData¥cC1021.c-2127)
関数 rcUnLoad() が異常終了しました。(psGetData¥rcUnLoad.c-71)
所要時間（秒）                :         1.60
異常終了しました。(mtu.c-1004)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

このエラーはデータの出力先ディスクに十分な空き領域がない場合に発生します。十分な領域を確保してから再試行して下さい。

[Query] ORA-01008: バインドされていない変数があります。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86> query
Multi-threaded Unloader version 4.2
                  :                                  :
                  「12.1.表示項目の共通部分(100頁」参照
                  :                                  :
任意問い合わせファイルを 1 個読込みました。(QUERYFIX) qryfix¥*.sql
任意問い合わせファイルを 2 個読込みました。(QUERYVAR) qryvar¥*.sql
バインド変数定義ファイルを読込みます。input¥bind.dat
:b_sal                        :         1,300
:b_hiredate                   :1980-01-01
複数の接続を確立します。要求された並列度は 4 です。
データ・ファイルと制御ファイルを出力中です。output
TableName                     :      NumRows:  Elapse:Rows/s:Bytes: kB/s:   Len
DEPT                          :            0:    0.00
ORA-01008: バインドされていない変数があります。
関数 rcStmtExecute() が異常終了しました。(cOciStmt.c-466)
関数 rcExecDefineAndFetch() が異常終了しました。(cOciStmt.c-980)
/* サンプルスクリプト（２）*/
/* 機能：バインド変数で与えられた条件を満たす行を戻す */
select *
from emp
where job = :b_job
and sal > :b_sal
and hiredate > to_date(:b_hiredate, 'yyyy-mm-dd')

SQL*Loader のコマンド行作成   :            3
DEPT                          :            4:    0.02:   197: 128.:    6:   32. CAN
Synchronized.
Error - １つ以上のスレッド関数が異常修了しました。
関数 rcSynchronizeSvc() が異常終了しました。(cOciSvc.c-1414)
関数 rcPostRepGetData() が異常終了しました。(psGetData¥cC1021.c-2016)
Waiting.
Synchronized.
```

```
Error - １つ以上のスレッド関数が異常修了しました。
関数 rcSynchronizeSvc() が異常終了しました。(cOciSvc.c-1414)
関数 rcForEachExList() が異常終了しました。(cList.c-397)
関数 rcGetData() が異常終了しました。(psGetData¥cC1021.c-2127)
関数 rcQuery() が異常終了しました。(psGetData¥rcQuery.c-34)
出力バイト総数 (kbytes)        :          0.44
所要時間 (秒)                  :          0.59
処理速度 (kbytes/秒)           :          0.75
異常終了しました。(mtu.c-1004)

C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

このエラーは Query 機能使用時に、半角コロン「:」で始まるプレースフォルダ文字列を含むSQL文に対して変数がバインドされていない場合に発生します。環境変数 INPUT の示す場所に、環境変数 FILEBIND（80 頁）の示す名前のテキストファイルを作成し、その内部にバインド変数名とその値を key=value 形式で記述してから再実行してください。

環境変数 SHOWSTMT（95 頁）を予め TRUE へセットしておくとどのクエリに対するバインド変数が不足しているのかが分かります。

[GetMeta] ORA-01436: ユーザー・データで CONNECT BY のループが発生しました。

このエラーは GetMeta 機能使用時に、スキーマに含まれている参照整合性制約の自己参照か循環参照を検出すると発生します。循環参照を構成する外部参照のひとつを明示的に無いものとみなして処理するよう設定する必要が有ります。プログラムフォルダ（43 頁）にある DelCircular.sql を編集してこの設定を行った後再実行してください。GetTblN 機能を実行して取得できるテキストファイルを使うと循環参照を見つけることが出来ます。

「循環参照を検出した為、再帰的検索を打切りました。」という部分がその場所です。

その他のアプローチとしては環境変数 DEPENDENT（77 頁）に値をセットしない方法もあります。

[UnLoad] ORA-01466: データを読み込めません - 表定義が変更されました

このエラーは読み込もうとしている表が truncate された後にフラッシュバック問合せが実行された時に発生します。

[Instant Client] ORA-12705: Cannot access NLS data files or invalid environment specified または ORA-12705: 無効または不明な NLS パラメータ値が指定されました。

Oracle Instant Client 利用時に環境変数の設定が不十分な場合に発生します。

環境変数 NLS_LANG を下記の事例に倣って設定し、再試行してください。

```
set NLS_LANG=japanese_japan.ja16sjis
```

27.3. 実行時の警告メッセージ

実行時に発生しやすい警告の実例と対処法について説明します。

Warning – 1 秒後にリトライします

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>unload
Multi-threaded Unloader version 4.2
                   :                                  :
                   「12.1.表示項目の共通部分(100頁」参照
                   :                                  :

ORA-00054: リソース・ビジー、NOWAITが指定されていました。
現在リトライ数 0 最大リトライ数 13 現在日付時刻 Sat Aug 23 16:48:16 2008
Warning - 1 秒後にリトライします。lock table "SYSTEM"."APLDDT01" in share mode nowait


ORA-00054: リソース・ビジー、NOWAITが指定されていました。
現在リトライ数 1 最大リトライ数 13 現在日付時刻 Sat Aug 23 16:48:17 2008
Warning - 2 秒後にリトライします。lock table "SYSTEM"."APLDDT01" in share mode nowait


ORA-00054: リソース・ビジー、NOWAITが指定されていました。
現在リトライ数 2 最大リトライ数 13 現在日付時刻 Sat Aug 23 16:48:19 2008
Warning - 4 秒後にリトライします。lock table "SYSTEM"."APLDDT01" in share mode nowait


ORA-00054: リソース・ビジー、NOWAITが指定されていました。
現在リトライ数 3 最大リトライ数 13 現在日付時刻 Sat Aug 23 16:48:23 2008
Warning - 8 秒後にリトライします。lock table "SYSTEM"."APLDDT01" in share mode nowait
```

この警告は製品プログラムが内部的に使用する作業表を指定のモードでロック出来なかった時に発生します。主な原因としては他の端末による接続でマルチ・スレッド・アンローダーが実行されている場合に起こります。他の端末による処理が終了すれば自動的に処理を再開します。

他の端末の処理が一向に終わらない場合はリトライを13回まで繰り返し、それでもロックできなかった場合には「Error – テーブル XXX のロック(%s)に失敗しました。他の端末が使用中です。」というメッセージを戻して異常終了します。

[ConsEnable] Warning – 有効化できません。(ORA-xxxxx)

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>ConsEnable
Multi-threaded Unloader version 4.2
                   :                                  :
                   「12.1.表示項目の共通部分(100頁」参照
                   :                                  :
整合性制約情報-Z               :           22:    0.00
整合性制約情報-A               :           22:    0.03
Warning - 有効化できません。(ORA-02437) PK_DEPT                         (a)
Warning - 有効化できません。(ORA-02270) FK_DEPTNO : DEPT : PK_DEPT : P     (b)
       8 個の主キー／一意性制約を有効化しました。
       0 個の参照制約を有効化しました。
       0 個の検査制約を有効化しました。
       2 個のデータベース・トリガーを有効化しました。
ラウンドトリップ（回）          :           43
所要時間（秒）                 :         2.07
正常終了しました。
```

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>
```

この警告は整合性制約の有効化に失敗したときに発生します。

a.　主キー／一意キー制約の場合

- 有効化に失敗した整合性制約の名前

b.　参照整合性制約の場合

- 有効化に失敗した整合性制約の名前
- 参照先表
- 参照先キー制約の名前
- 参照先キー制約の型(P:主キー制約、U:一意キー制約の何れか)

が順に示されます。EXCEPTIONS 表（例外表）へ制約違反の原因となった行の ROWID が格納されるので、これを基に違反の原因を取り除いた後、再試行して下さい。

```
select * from exceptions where owner = ＜環境変数 SRC_USER＞;
```

で行を取り出すことが出来ます。

Warning - 戻された LONG 型/LONG RAW 型値はホスト変数の最大長を超えたため切捨てられました。

```
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x86>unload
Multi-threaded Unloader version 4.2
                    :                                    :
                    「12.1.表示項目の共通部分(100頁」参照
                    :                                    :
表に含まれている行数の測定      :             1:    0.00
複数の接続を確立します。要求された並列度は 4 です。
整合性制約情報-Z               :            12:    0.00
整合性制約情報-B               :             5:    0.04
出力対象の表名                 :             1:    0.46
複数の接続を確立します。要求された並列度は 4 です。
データ・ファイルと制御ファイルを出力中です。output
TableName                      :       NumRows:  Elapse:Rows/s:Bytes: kB/s:   Len
Not NULL constraints           :             5:    0.00
セグメント属性                 :     Executing:    0.67
ORA-24345: 切捨てまたは NULL フェッチ・エラーが発生しました。(cOciStmt.c-748)
Warning - 戻された LONG 型/LONG RAW 型値はホスト変数の最大長を超えたため切捨てられました。
環境変数 MAXLONGSIZE へ必要十分な値を設定して下さい。現在値 1,000       (a)
表示項目: TABLE_NAME.COLUMN_NAME, ROW_COUNT, SIZE
IDL_UB1$.PIECE, 2, ==4,870                                               (b)
IDL_UB1$.PIECE, 4, ==12,288
IDL_UB1$.PIECE, 6, ==1,132
IDL_UB1$.PIECE, 8, ==2,982
IDL_UB1$.PIECE, 11, ==3,540
```

```
IDL_UB1$.PIECE, 13, ==12,288
```

この警告は出力対象となった表（事例ではIDL_UB1$と表示）から取り出されたLONG型あるいはLONG RAW型のデータ長が大きすぎる為、バッファに収まらなかったときに発生します。必要なデータ長は(b)に示されます。

ただしここに表示される数値は符号付き16ビット整数型(OCIInd型)の変数から取り出されたものなので表現可能な最大値は65,535までとなります。また現在のバッファのサイズは(a)に示されます。事例では環境変数MAXLONGSIZE（85頁）が1000に指定されていたのでそのサイズを超えたデータ格納されていた2, 4, 6, … 23行目の行取り出しで警告が戻されています。

バッファのサイズを環境変数MAXLONGSIZEによって調節した後、再度実行して下さい。このとき、環境変数BULK_SIZE（75頁）とMAXLONGSIZEを乗じたバイト数＋αのメモリが実行時に確保されます。ただしBIND_SIZE（75頁）へ指定したサイズを超える事はありません。

27.4. SQL*Loader実行時のメッセージ

ORA-02359: データ・ファイルのフィールドが最大長を超えました。

```
レコードxx: 拒否されました。- 表"xxxxxxxxxxxxx",列"xxxxxxxx"でエラーが発生しました。
マルチバイト・キャラクタでエラーが発生しました。
```

全角文字が格納されている列で列値の最後の文字が全角文字の1バイト目である時発生します。エラーが生じた行データだけ<表名>.badファイルに控えが出力されているのでこれを使い、原因となる列値を修正した後再度ローディングを試みてください。

SQL*Loader-510: データ・ファイル(xxx.dat)の物理レコードが最大長(xxx)を超えています。または SQL*Loader-555: 認識できない処理オプションです。

このエラーは特大のLONG/LONG RAW型のロード時に発生します。引数としてreadsizeへ、より大きな値をを明示的に与えて下さい（省略時初期値は1048576バイト）。この引数は_par0.txt, _par1.txtで指定されています。

ORA-06564: オブジェクトALIAS1は存在しません。

外部表を定義するスクリプトを（環境変数GETEXTTBL（83頁）を設定した時に出力される）実行した時に上記のエラーメッセージが戻される時は下記の事例に倣ってディレクトリ・オブジェクトを事前に作成してから再試行してください。

（想定事例：systemユーザにてディレクトリ・オブジェクト'ALIAS1'を作成し、それを使用する権限をSCOTTへ付与。その後、SCOTTで再接続しスクリプトを実行。）

```
connect system/manager
create or replace directory ALIAS1 as 'C:¥borland¥USER¥mtu¥output';
（上記 SQL は CREATE ANY DIRECTORY システム権限が必要です）
grant read, write on directory ALIAS1 to scott;
connect scott/tiger
create table "外部表名"
.....
```

```
) ORGANIZATION EXTERNAL (
TYPE ORACLE_LOADER
DEFAULT DIRECTORY "ALIAS1"
.....
);
```

27.5. その他の異常終了

製品の v4.16 以上をお使いの場合、このセクションに表記した機能を使って異常終了を診断するための情報を得ることが出来ます。それ未満のバージョンの場合は、以前のバージョンに付属する取扱説明書の「トラブルシューティング」をご覧下さい。

27.5.1. クラッシュ

製品がクラッシュすると自動的にプロセスダンプファイルが出力されます。画面出力や、トレースファイル(環境変数 FILETRACE、83 頁)と共に、プロセスダンプファイルを製品サポート係へお送りください。製品サポートにはサポート契約が必要です。

プロセスダンプファイルのファイル名は次の形式で与えられます。

```
環境変数 OUTPUT へ指定した値のフォルダ名 +"_mtu"+"_(半角数字 5 桁のプロセス ID)"+"(半角数字 2
桁の連続番号)"+".dmp"
```

また画面上に次のように表示され、プロセスダンプが取得されたことが通知されます。

```
Error - SEH caught an exception. (0xc00000fd: EXCEPTION_STACK_OVERFLOW)
C:¥ProgramData¥PLUMSIX¥mtu_4.2¥x64¥output¥_mtu_05432_01.dmp
Mini-dump file was saved.
```

27.5.2. ハングアップ

ハングアップの場合は、Oracle の動的性能表等で mtu.exe プロセスによるセッションが、アイドル待機イベントで待機していることを確認の上、CTRL+Break を押下して下さい。前項で示した場所と同じ場所にプロセスダンプを出力することが出来ます。

画面出力や、トレースファイル()と共に、プロセスダンプファイルを製品サポート係へお送りください。

28. 応用事例

UnLoad/Query 機能を使って得たSQL*Loader 用制御ファイルを適宜編集することにより、テーブルの仕様変更に伴う列の追加や Not Null 制約の追加、列の削除に対して柔軟に対処できます。このセクションでは応用例を示します。

変更前の状態の制御ファイルについては「29.出力サンプル(186 頁)」をご覧下さい。

28.1. 列の追加

事例:EMP 表に社員の電子メールアドレスを格納するための列 EMAIL が追加になった。この列に Not Null 制約が施行される。

```
LOAD INFILE "EMP" "VAR 10"
INTO TABLE "EMP"
TRUNCATE REENABLE FIELDS TERMINATED BY ','
(  "EMPNO"                          DECIMAL EXTERNAL(5)
,  "ENAME"                          CHAR(10) ENCLOSED BY '"'
,  "JOB"                            CHAR(9) ENCLOSED BY '"'
,  "MGR"                            DECIMAL EXTERNAL(5)
,  "HIREDATE"                       DATE(16) "YYYYMMDDHH24MISS" ENCLOSED BY '"'
,  "SAL"                            DECIMAL EXTERNAL(9)
,  "COMM"                           DECIMAL EXTERNAL(9)
,  "DEPTNO"                         DECIMAL EXTERNAL(3)
,  "EMAIL"     CONSTANT 'dummy@hq.acme.com'                        (a)
)
```

(a)に示すように、制御ファイルに CONSTANT 句オプションを含め、追加した列へ仮の値を導入すれば Not Null 制約違反が発生することを抑止できます。追加した列が可変長データ型である場合、実際に代入されることになる値のデータ長よりも長い仮の値をすると行移行の発生を抑えることができます。

28.2. Not Null 制約の追加

事例:EMP 表の JOB 列に Not Null 制約が追加された。

```
LOAD INFILE "EMP" "VAR 10"
INTO TABLE "EMP"
TRUNCATE REENABLE FIELDS TERMINATED BY ','
(  "EMPNO"                          DECIMAL EXTERNAL(5)
,  "ENAME"                          CHAR(10) ENCLOSED BY '"'
,  "JOB"                            CHAR(9) ENCLOSED BY '"'
                                       "nvl(:¥"JOB¥",'dummy')"        (a)
,  "MGR"                            DECIMAL EXTERNAL(5)
```

```
,   "HIREDATE"                          DATE(16) "YYYYMMDDHH24MISS" ENCLOSED BY '"'
,   "SAL"                               DECIMAL EXTERNAL(9)
,   "COMM"                              DECIMAL EXTERNAL(9)
,   "DEPTNO"                            DECIMAL EXTERNAL(3)
)
```

(a)に示すように、ENCLOSED BY 句オプションの後に2重引用符で囲まれた SQL 関数を含む式を記述すると演算結果を列値として代入できます[33]。この例では JOB IS NULL となった場合、仮の文字列値'dummy'を強制的に代入しています。このオプションはダイレクト・パス・ロードでは使用できません。

28.3. 列の削除

事例:EMP 表の COMM 列が削除された。

```
LOAD INFILE "EMP" "VAR 10"
INTO TABLE "EMP"
TRUNCATE REENABLE FIELDS TERMINATED BY ','
(  "EMPNO"                              DECIMAL EXTERNAL(5)
,   "ENAME"                             CHAR(10) ENCLOSED BY '"'
,   "JOB"                               CHAR(9) ENCLOSED BY '"'
,   "MGR"                               DECIMAL EXTERNAL(5)
,   "HIREDATE"                          DATE(16) "YYYYMMDDHH24MISS" ENCLOSED BY '"'
,   "SAL"                               DECIMAL EXTERNAL(9)
,   "COMM"          FILLER              DECIMAL EXTERNAL(9)                 (a)
,   "DEPTNO"                            DECIMAL EXTERNAL(3)
)
```

(a)に示すように、FILLER 句オプションを使用するとデータ・ファイルから読み取った列値を表の列値へ代入する操作をスキップすることが出来ます。なお、FILLER は Oracle8i から使用可能になったオプションです。

[33] R8.1.6/R8.1.5 の SQL*Loader では式の指定でエスケープ文字を含められないので「nvl(:JOB,'dummy')」と書換えてください。

29. 出力サンプル

この章では MTU が出力するファイルのサンプルを提示します。サンプルでは「EMP」表を使用します。「EMP」表は Oracle のデモンストレーション用に用意されているスクリプトで作成することの出来る、SCOTT スキーマに含まれるオブジェクトのひとつです。

下記の場所に SCOTT スキーマの作成スクリプトが有りますのでお客様自身のオラクルデータベースへこの表をインストールして確認することが可能です。

- ［Oracle 8i 未満］

```
Windows プラットホーム      @%ORACLE_HOME%¥rdbms¥admin¥utlsampl.sql
UNIX プラットホーム         @${ORACLE_HOME}/rdbms/admin/utlsampl.sql
```

- ［Oracle 8i 以上］

```
プラットホーム共通          @?/rdbms/admin/utlsampl.sql
```

29.1. UnLoad 機能（CSV 形式）

環境変数を出荷時設定のままで使用し、EMP 表を UnLoad すると下記サンプルのような出力結果が得られます。

29.1.1. 制御ファイル

```
LOAD INFILE "EMP" "VAR 10"
INTO TABLE "EMP"
TRUNCATE REENABLE FIELDS TERMINATED BY ','
(  "EMPNO"                              DECIMAL EXTERNAL(5)
,  "ENAME"                              CHAR(10) ENCLOSED BY '~~'
,  "JOB"                                CHAR(9) ENCLOSED BY '~~'
,  "MGR"                                DECIMAL EXTERNAL(5)
,  "HIREDATE"                           DATE(14) "YYYYMMDDHH24MISS" ENCLOSED BY '~~'
,  "SAL"                                DECIMAL EXTERNAL(9)
,  "COMM"                               DECIMAL EXTERNAL(9)
,  "DEPTNO"                             DECIMAL EXTERNAL(3)
)
```

29.1.2. データ・ファイル

```
00000000597369,~~SMITH~~,~~CLERK~~,7902,~~19801217000000~~,800,,20,
00000000667499,~~ALLEN~~,~~SALESMAN~~,7698,~~19810220000000~~,1600,300,30,
00000000657521,~~WARD~~,~~SALESMAN~~,7698,~~19810222000000~~,1250,500,30,
```

```
00000000627566,~~JONES~~,~~MANAGER~~,7839,~~19810402000000~~,2975,,20,
00000000687654,~~MARTIN~~,~~SALESMAN~~,7698,~~19810928000000~~,1250,1400,30,
00000000627698,~~BLAKE~~,~~MANAGER~~,7839,~~19810501000000~~,2850,,30,
00000000627782,~~CLARK~~,~~MANAGER~~,7839,~~19810609000000~~,2450,,10,
00000000627788,~~SCOTT~~,~~ANALYST~~,7566,~~19870419000000~~,3000,,20,
00000000597839,~~KING~~,~~PRESIDENT~~,,~~19811117000000~~,5000,,10,
00000000657844,~~TURNER~~,~~SALESMAN~~,7698,~~19810908000000~~,1500,0,30,
00000000607876,~~ADAMS~~,~~CLERK~~,7788,~~19870523000000~~,1100,,20,
00000000597900,~~JAMES~~,~~CLERK~~,7698,~~19811203000000~~,950,,30,
00000000617902,~~FORD~~,~~ANALYST~~,7566,~~19811203000000~~,3000,,20,
00000000617934,~~MILLER~~,~~CLERK~~,7782,~~19820123000000~~,1300,,10,
```

29.2. UnLoad 機能(固定長形式)

環境変数を「表 34 固定長データ形式の設定例」のように変更し、EMP 表を UnLoad すると下記サンプルのような出力結果が得られます。環境変数 LISTFIXED(84 頁)、FILEFIXED(81 頁)へ指定されなかったその他の表は CSV 形式で出力されます。

固定長形式で出力する場合、半角空白文字列をデータとして扱うときの問題点があります。詳細は「6.6. 固定長形式に含まれる空白文字(35 頁)」の記事を参照してください。

表 34 固定長データ形式の設定例

環境変数名	出荷時設定値	変更後の値
LISTFIXED(84 頁)、FILEFIXED(81 頁)	＜空＞	EMP

29.2.1. 制御ファイル

```
LOAD INFILE "EMP" "FIX 66"
INTO TABLE "EMP"
TRUNCATE REENABLE FIELDS
(  "EMPNO"                          POSITION( 1: 5) DECIMAL EXTERNAL
,  "ENAME"                          POSITION( 6:15) CHAR
,  "JOB"                            POSITION(16:24) CHAR
,  "MGR"                            POSITION(25:29) DECIMAL EXTERNAL
,  "HIREDATE"                       POSITION(30:43) DATE "YYYYMMDDHH24MISS" NULLIF
"HIREDATE"=BLANKS
,  "SAL"                            POSITION(44:52) DECIMAL EXTERNAL
,  "COMM"                           POSITION(53:61) DECIMAL EXTERNAL
,  "DEPTNO"                         POSITION(62:64) DECIMAL EXTERNAL
)
```

29.2.2. データ・ファイル

```
7369SMITH     CLERK      790219801217000000   800           20
7499ALLEN     SALESMAN   769819810220000000  1600     300   30
7521WARD      SALESMAN   769819810222000000  1250     500   30
7566JONES     MANAGER    783919810402000000  2975           20
7654MARTIN    SALESMAN   769819810928000000  1250    1400   30
7698BLAKE     MANAGER    783919810501000000  2850           30
7782CLARK     MANAGER    783919810609000000  2450           10
7788SCOTT     ANALYST    756619870419000000  3000           20
7839KING      PRESIDENT      19811117000000  5000           10
7844TURNER    SALESMAN   769819810908000000  1500       0   30
7876ADAMS     CLERK      778819870523000000  1100           20
7900JAMES     CLERK      769819811203000000   950           30
7902FORD      ANALYST    756619811203000000  3000           20
7934MILLER    CLERK      778219820123000000  1300           10
```

29.3. UnLoad 機能 - 外部表

外部表の定義は環境変数 GETEXTTBL(83 頁)へ任意文字列を設定した時に出力することが出来ます。

(a)にあるディレクトリ・オブジェクトはこの外部表を定義するスクリプトを実行する前に作成し、read/write 権限を実行するユーザへ与えておく必要があります。ディレクトリ・オブジェクトの名前は環境変数 DIRALIAS(77 頁)で変更することが出来ます。

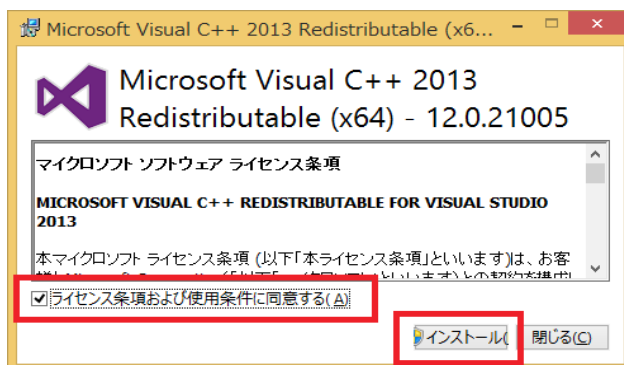
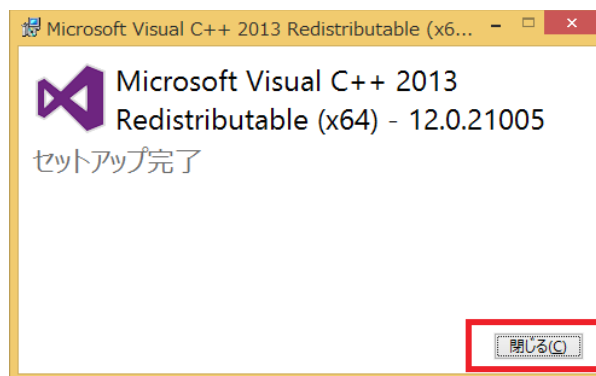
```
-- drop table "EMP";
create table "EMP"
(  "EMPNO"                          number(4)
,  "ENAME"                          varchar2(10)
,  "JOB"                            varchar2(9)
,  "MGR"                            number(4)
,  "HIREDATE"                       date
,  "SAL"                            number(7,2)
,  "COMM"                           number(7,2)
,  "DEPTNO"                         number(2)
) ORGANIZATION EXTERNAL (
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY "ALIAS1"                          (a)
  ACCESS PARAMETERS (
    RECORDS VARIABLE 10
    BADFILE "EMP"
    LOGFILE "EMP"
    FIELDS TERMINATED BY ','
    (  "EMPNO"                        CHAR(5)
    ,  "ENAME"                        CHAR(10) ENCLOSED BY '~~'
    ,  "JOB"                          CHAR(9) ENCLOSED BY '~~'
    ,  "MGR"                          CHAR(5)
    ,  "HIREDATE"                     DATE(14) "YYYYMMDDHH24MISS" ENCLOSED BY '~~'
    ,  "SAL"                          CHAR(9)
```

```
    ,  "COMM"                             CHAR(9)
    ,  "DEPTNO"                           CHAR(3)
    )
  ) LOCATION ('EMP.dat')
);
```

29.4. GetMeta 機能

```
/*
 * Tabs: 000025. EMP
 */
drop table "EMP";
create table "EMP"
( /* 0001 */ "EMPNO"                            number(4)
, /* 0002 */ "ENAME"                            varchar2(10)
, /* 0003 */ "JOB"                              varchar2(9)
, /* 0004 */ "MGR"                              number(4)
, /* 0005 */ "HIREDATE"                         date
, /* 0006 */ "SAL"                              number(7,2)
, /* 0007 */ "COMM"                             number(7,2)
, /* 0008 */ "DEPTNO"
, constraint "PK_EMP"                           primary key
  ( /* 0001 */ "EMPNO"
  ) not deferrable initially immediate disable novalidate
, constraint "FK_DEPTNO"                        foreign key
  ( /* 0008 */ "DEPTNO"
  ) references "DEPT"
  ( /* 0001 */ "DEPTNO"
  ) not deferrable initially immediate disable novalidate
) pctfree 10 pctused 40 initrans 1 maxtrans 255
  nocache parallel (degree 1 instances 1) tablespace "SYSTEM"
  storage (initial 64k minextents 1 maxextents unlimited
  freelists 1 freelist groups 1 buffer_pool DEFAULT)
;
```

30. Visual Studio 2013 の Visual C++ 再頒布可能 パッケージの導入

- パッケージを「表 7:再頒布可能パッケージの入手先（31 頁）」に示したサイトから入手して下さい。
- ダウンロードしたパッケージの実行形式をダブルクリックしてください。
- 次のようなダイアログボックスが表示されます。「ライセンス条項および使用条件に同意する」をチェックの上、インストールボタンを押してください。

- インストールが終わると次のダイアログボックスが表示されます。完了ボタンを押してください。

- インストールが完了すると、「コントロールパネル」の「プログラムと機能」でパッケージ名を確認することができるようになります。

Microsoft Visual C++ 2013 Redistributable (xNN) – 12.0 という名前で確認することが出来ます。NN には 86 か 64 の何れかが表示されます。

31. 製品フォルダのコピー方法

このセクションでは、製品フォルダのコピー方法について説明します。製品フォルダは、製品をインストール時に固有の場所へ作成され「表 11:製品フォルダとプログラムフォルダ(43 頁)」、場所を変更する事は出来ません。

都合により、製品フォルダにあるスクリプト・ファイルを別なフォルダ位置で使用したい場合、製品フォルダをコピーする事が可能です。コピーしたフォルダについて自由な名前を与えたり、スクリプト・ファイルを自由に編集するることも可能です。また製品フォルダ上のファイルと異なり、製品アンインストール時にも削除されることがありません。

コピー方法について以下にその手順を説明します。

- 下記のように操作してコマンド・プロンプトを表示させて下さい。

「スタート」→「すべてのプログラム」→「PLUMSX」→「mtu v4.2 ??-bit 有償版」→「コマンド・プロンプト」

- 次のコマンドを実行して、製品フォルダを任意の場所へコピーして下さい。"C:¥TEMP¥MTU_HOME"と指定した部分は、お客様の希望する任意のフォルダ名に差し替える事が出来ます。

```
robocopy . C:¥TEMP¥MTU_HOME /E
```

- output フォルダのコピーが不要な場合は次のように入力します。この場合、コピー先の Env.bat 内で指定している環境変数 OUTPUT の値も実在のフォルダを指し示すよう編集しなおす事を忘れないで下さい。

```
robocopy . C:¥TEMP¥MTU_HOME /E /XD output
```

32. セット内容

配布用のインストール・パッケージ・ファイル「mtu_xxx.msi（xxx は bit 数によって異なる番号）」には以下のファイルが含まれています。インストール先の「プログラム・フォルダ」と「製品フォルダ」に関する説明は表 11:製品フォルダとプログラムフォルダ（43 頁）を参照してください。

表 35 プログラム・フォルダへインストールされるもの

ファイル名	説明
DelCircular.sql	自己参照・循環参照の FK に関する行を削除するスクリプト
mkcrd.exe	資格証明情報作成ユーティリティー
mpx.exe	多重化プロセス実行ユーティリティー製品プログラム本体 (x64 / x86 用)
MTU.xla	クイック・エクスポート用 Excel アドイン
mtu.exe	マルチ・スレッド・アンローダー 製品プログラム本体 (x64 / x86 用)
sqlite3.dll	SQLite ver.3 共有ライブラリ
sqlite3.exe	SQLite ver.3 コマンドライン・ユーティリティー
PsPas.cer	ライセンス認証用公開鍵
PsPasCli.dll	ライセンス認証用クライアントモジュール
PsPasCom.dll	ライセンス認証用共通モジュール
UnRegister.exe	ライセンス認証登録抹消ユーティリティー
doc¥Intro_MTU.pdf	マルチ・スレッド・アンローダー製品紹介
doc¥license.rtf	使用許諾契約書
doc¥readme_mpx.pdf	プロセス多重化実行ユーティリティー取扱説明書
doc¥readme_mtu.pdf	マルチ・スレッド・アンローダー取扱説明書 （このファイル）
doc¥release.txt	リリース・ノート (readme.txt を読む前にお読みください。)
doc¥WP_NewFeature_v40_All.pdf	新機能説明書
sample¥demo_vpd.sql	Virtual Private Database 設定例のスクリプト
sample¥GetClobVal.sql	任意問合せ例 – その 4 XMLType 型を含む表の問合せ
sample¥GetSchema.sql	任意問合せ例 – その 5 dbms_metadata パッケージ使用例
sample¥LineCount.pl	テキスト行数測定スクリプトファイル（逐次処理）
sample¥LineCountP.pl	テキスト行数測定スクリプトファイル（並列処理）
sample¥NamedPipeMulti.cpp	C++/マルチスレッド版名前つきパイプ利用プログラムサンプル
sample¥NamedPipeMulti.java	java/マルチスレッド版名前つきパイプ利用プログラムサンプル
sample¥NamedPipeSingle.cpp	C++/シングルスレッド版名前つきパイプ利用プログラムサンプル
sample¥NamedPipeSingle.java	java/シングルスレッド版名前つきパイプ利用プログラムサンプル
sample¥RebuildSequences.sql	順序オブジェクトを再定義する為のスクリプトを作成するスクリプト
sample¥seldom_use.sql	あまり使われることの無いデータ型を含む表を定義するサンプル

表 36:製品フォルダへインストールされるもの

ファイル名	説明
ConsDisable.bat	整合性制約の無効化機能を利用する為のスクリプト
ConsEnable.bat	整合性制約の有効化機能を利用する為のスクリプト
env.bat	環境変数を設定する為のスクリプト
GetFkrb.bat	外部キー再構成スクリプトファイル出力スクリプト
GetMeta.bat	スキーマオブジェクト定義 SQL 文出力スクリプト
GetPlsN.bat	名前付き PL/SQL 間参照階層図（逆展開）出力スクリプト
GetPlsP.bat	名前付き PL/SQL 間参照階層図（正展開）出力スクリプト
GetTblN.bat	表間参照階層図（逆展開）出力スクリプト
GetTblP.bat	表間参照階層図（正展開）出力スクリプト
Install.bat	製品コンポーネントのインストールをする為のスクリプト
mtu_prompt.bat	MTU のコマンドプロンプト用スクリプト
Platform.bat	プラットフォーム依存の環境変数設定スクリプト
QikExp.bat	Microsoft Excel 用のデータを手軽に出力するためのスクリプト
QikExp.dat	QikExp.bat で使用される省略時初期値データ
Query.bat	任意問合せファイル出力機能を利用する為のスクリプト
ReLoad.bat	多重化再ロード処理を実行するためのスクリプト
UnInstall.bat	製品コンポーネントのアンインストールをする為のスクリプト
UnLoad.bat	表データ・ファイル出力機能を利用する為のスクリプト
input¥_cre_pub_syn.sql	アンインストール用スクリプト例（Install 時に書き換えられる）
input¥_drp_prod.sql	アンインストール用スクリプト例（Install 時に書き換えられる）
input¥_drp_pub_syn.sql	アンインストール用スクリプト例（Install 時に書き換えられる）
input¥bind.dat	バインド変数定義ファイルの例
input¥drop_all_apldd.sql	MTU 用オブジェクトの強制一括削除スクリプト
input¥excpt.dat	出力除外表名リストファイル
input¥fixed.dat	出力対象表名リストファイル（固定長データ形式）
input¥PostRepPls.sql	データ問合せを反復した後に実行する PL/SQL スクリプト
input¥PreRepPls.sql	データ問合せを反復した前に実行する PL/SQL スクリプト
input¥table.dat	出力対象表名リストファイル（CSVデータ形式）
qryfix¥dept.sql	任意問合せ例 – その 1 固定長データ形式
qryfix¥UNION_EMP_A.sql	集合化クエリの例－その1
qryfix¥UNION_EMP_B.sql	集合化クエリの例－その2
qryvar¥emp.sql	任意問合せ例 – その 2 CSVデータ形式
qryvar¥sal_avg.sql	任意問合せ例 – その 3 CSVデータ形式

33. 製品に関する問合せ先

- 受付時間

  年末年始、土曜日、日曜日、祝日を除く毎日。午前 9 から午後 6 時まで。尚、受付時間の終了間際にお問合せいただいた場合は、ご回答が次の受付時間まで遅れることがございますのであらかじめご了承ください。

- 製品に関するお見積、ご導入前の相談、評価ライセンス発行のお申込、ご契約、パブリシティーに関するお問合せは下記までご連絡願います。

株式会社プラムシックス
営業部 MTU 担当
電話 043-223-5202
メール info@plumsix.co.jp

- ライセンス認証、製品の使い方、仕様、障害に関する問合せは下記までご連絡願います。但し、ライセンス・キーお受け取り後 90 日以内、または保守サポート契約締結済みのお客様に限らせて頂きます。尚、お問い合わせを受け付ける際にサポート契約番号を確認させていただく場合がございます。

株式会社プラムシックス
製品サポート係
電話 043-223-5206
メール mtu_support@plumsix.co.jp

34. 商標について

マルチ・スレッド・アンローダー、および Multi-threaded Unloader は株式会社プラムシックスの登録商標です。

オラクル、および ORACLE は米国 Oracle Corporation の登録商標です。

Microsoft、Visual Studio、Windows、Windows Server、Windows XP、Windows Vista、Visual C++ は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

Intel、Pentium、および Itanium は、米国およびその他の国における、Intel Corporation またはその子会社の商標または登録商標です。

他のブランドおよび製品名は、各社の商標および登録商標です。